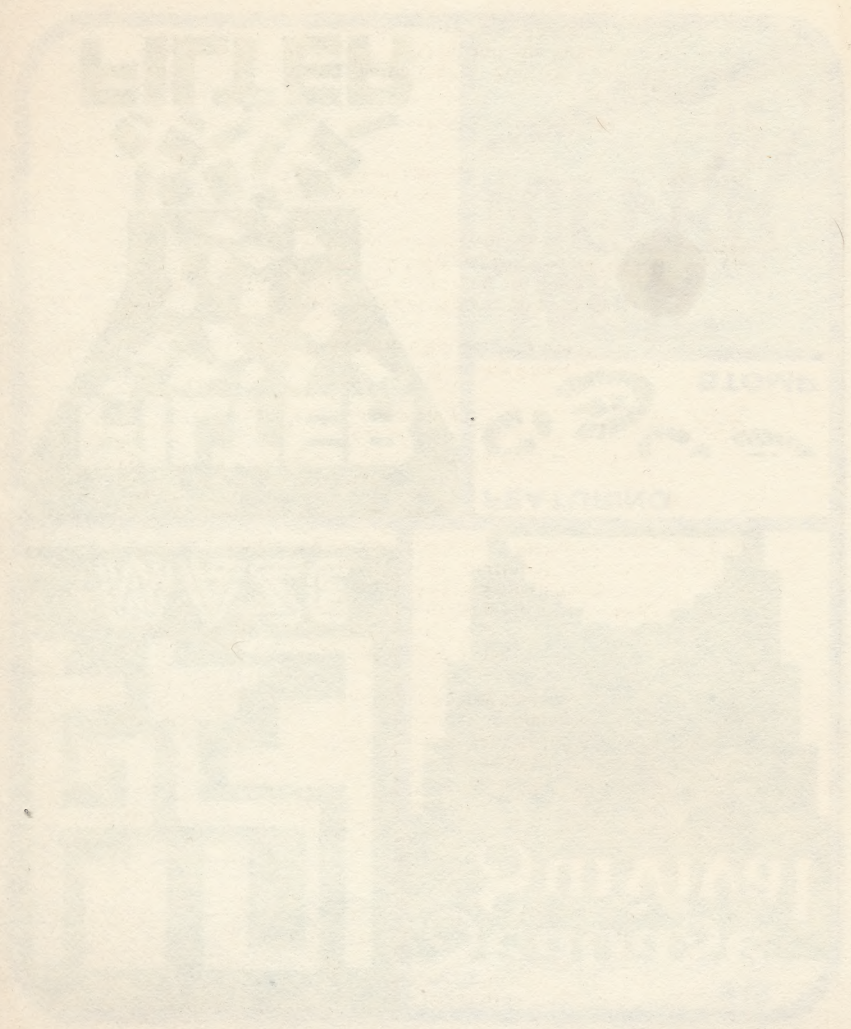


SoftSide[®] Selections



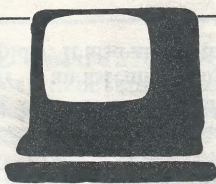
A product of **SoftSide** Publications, Inc.
10 Northern Blvd., Amherst, NH 03031

1870-1871
1871-1872
1872-1873



26/6ction2

01/1/196



SoftSide® Selections

TRS-80® is a trademark of the Tandy Corporation.

Index

STOMP

by Rich Bouchard

Here is **SoftSide's** latest assault on typos, to make life easier for users who type in their programs from printed listings. This new, official debugging utility replaces SWAT, and offers still more assistance if you type in BASIC code

2

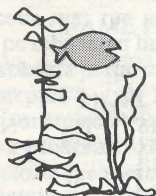


Seabase Survival

by Michael P. McEwen

Your mission, should you choose to accept it, is to save Atlantis II, a seabase resting on the ocean floor, while your technical crew develops new technology to compensate for the dying geothermal energy source.....

7



DV BONUS PROGRAM: Filter

by Michael J. Sorens

Drive your printer to generate stylish output with this program. Add text interactively while printing, and check pertinent document statistics, all with one or two keystrokes

14



Maze

by John F. Gabriel Jr.

Create an almost infinite number of mazes, and wend your way through them — from a mouse-eye point of view

24



Adventure Disk and Cassette Bonus

SoftSide Adventure Series: Jack the Ripper II

by Peter Kirsch

Suppose the evil Ripper never really died. This adventure casts you in the role of a modern Scotland Yard Inspector on the trail of Jack the Ripper number two.....

36



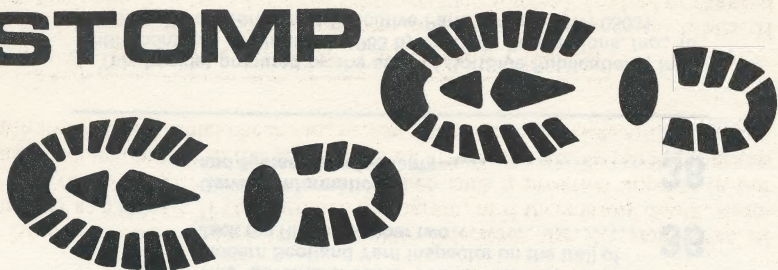
General Information

and System Requirements

38

This booklet prepared by the staff of SoftSide Publications, Inc.,
Entire contents copyright © 1983 by SoftSide Publications, Inc., 10
Northern Blvd., Northwood Executive Park, Amherst, NH 03031.

STOMP



by Kerry Shetline
translation by Rich Bouchard

STOMP is a utility to help track down typing errors in programs from *SoftSide Selections*. It replaces *SWAT* beginning with Issue 45.

Note: Please read *all* the instructions for *STOMP* before beginning.

Why *STOMP*?

STOMP (*Stop Typos On My Programs*) aids the reader who types BASIC programs from a listing in *SoftSide* or *SoftSide Selections*. It is *SoftSide's* new official debugging utility. Until now, this function was filled by *SWAT*, which readers hailed as a great aid in tracking down the inevitable errors that occur when you are typing several pages of BASIC code. The concise *SWAT* table narrows the search for errors to no more than twelve lines of code. For complex programs, you can reduce this number at will.

Nonetheless, *SWAT* has drawbacks and deficiencies. For one thing, to get a matching *SWAT* table, readers must type every line of the program exactly as it appears in the magazine. Even REM statements, which BASIC ignores, have to be replicated precisely. Also, *SWAT* can't detect simple transpositions. This means that the numbers 32767 and 36277 are identical to *SWAT*. If these numbers are part of a DATA statement for a machine language routine, the computer may hang up, or important data may vanish. Furthermore, because *SWAT* is written entirely in BASIC, it is quite slow. In sum, *SWAT*, although a big help, leaves much room for improvement.

STOMP corrects *SWAT's* deficiencies. It is faster, easier to use, and more reliable than *SWAT*. In addition, *STOMP* ignores REM statements and insignificant spaces. If you type BASIC programs from *SoftSide Selections*, *STOMP* will save you many hours.

You may omit any REM statements in our programs. If you do so, be sure to remove any colon (:) immediately preceding the word REM. In addition, feel free to add REM statements (within the constraints of memory) without changing the *STOMP* tables.

The tables *STOMP* generates are identical in appearance to *SWAT* tables, so *SWAT* veterans will have no trouble interpreting *STOMP* tables. Type *STOMP* in now, and start using it right away!

Starting Up With *STOMP*

STOMP is an assembly language program. This means that it generates its tables rapidly. It also means that you need append nothing to the program you

are testing. Type the *STOMP Generator* program listed below. Save it. Check it with *SWAT* to be sure it is correct, then run it. The *Generator* program does its own check to be sure you have entered it correctly.

If you see the message "Sorry, but there is an error in the DATA statements in lines 1000-1300," then carefully check these lines, correct the error or errors, save the program and try again.

Using *STOMP*

When you want to use *STOMP*, run the *Generator* program. It installs *STOMP* at the top of memory. Next, type and save the program you want to check. If you have already saved the program, you may, of course, simply load it.

STOMP uses the NAME command. To get a *STOMP* table, you may simply type

NAME

This command generates the type of *STOMP* table we usually publish. If you want a printout of your table, type

NAME P

When the *STOMP* table exceeds one screenful, the program pauses until you press Enter. This pause does not occur if you have the table printed.

For particularly complex programs, we publish expanded *STOMP* tables that can locate typing errors to within two or three lines. Such tables carry a label like, "Modified parameters: Line count = 3, Byte count = 200." To generate such tables, type

NAME lc,bc

where lc and bc represent the particular numbers for your table. To get a printout, use the command

NAME lc,bc,P

If your table matches the published one, you may begin using your program. If they do not, see the section entitled, "What's All This About Parameters?" which further explains parameters.

When you are done using *STOMP*, you will want to remove it to release the memory it occupies. The following commands do this.

NAME *:CLEAR 50

Note that some programs in *SoftSide Selections* use the same area of memory as *STOMP*. If you run such a program, or if there is any doubt, simply run the *Generator* program again. When such a program appears in our pages, we will note it in the accompanying article. Do not run such programs until the *STOMP* tables check out, or you must re-run the *Generator* program.

What To Do If The *STOMP* Tables Don't Match

- First examine the listed line numbers in the first columns of the tables. If they don't match, it probably means that you have inserted, omitted or changed

STOMP

STOMP

at least one line number. A large omission from a line can also throw off the first column. An extra or missing line affects all entries in the first column from that point on. Search the lines indicated by the first erroneous entry for the bad line, and correct it. Then try running *STOMP* again to see if the entries match.

● If, after you have verified all the line numbers, there are still discrepancies in the second or third column, you need to make a more detailed search.

(a) If the length for a range of lines is too big, it is likely that you have misspelled a BASIC keyword. Another possibility is that you typed too many spaces or other characters in a PRINT or DATA statement. Check all variable names.

(b) If the length entry is too small, check for omissions. Did you leave out part of a line? Did you miss a character in a PRINT or DATA statement?

(c) If the length is correct but the *STOMP* code is wrong, check the appropriate lines for transposed or mistyped characters.

Remember that *STOMP* ignores REM statements and spaces that aren't significant. Anything within quotation marks is vital, and you must type it precisely. The same goes for DATA statements, except for spaces between the word DATA and the first item, which are not important.

What's All This About Parameters?

Parameters are simple to understand. Normally, *STOMP* starts a new line in its table when it has checked twelve lines or 500 bytes of memory, whichever comes first. This means that it will narrow your search for typographical errors to about a twelve-line range of the program. If a program is particularly complex, we make *STOMP* generate new table entries at smaller intervals, making your search for typos still easier.

```

SS SS SS SS SS SS SS SS SS SS SS
SS                                     SS
SS      TRS-80 BASIC                SS
SS      'STOMP Generator'           SS
SS      Author: Rich Bouchard      SS
SS      Copyright © 1983           SS
SS      SoftSide Publications, Inc SS
SS                                     SS
SS SS SS SS SS SS SS SS SS SS SS

```

**If you don't wish to type this program, it is
available on Issue #45 SoftSide DV and CV.**

```

100 CLEAR50
110 CLS:PRINT"SOFTSIDE MAGAZINE PRESENTS... STOMP":PRINT
120 PRINT"YOU HAVE A ";
130 POKE-1,0:IFPEEK(-1)=0THENX=254:PRINT"48";ELSEPOKE-16385,0:IF
PEEK(-16385)=0THENX=190:PRINT"32";ELSEX=126:PRINT"16";
140 PRINT"K ";
150 POKE16561,255:POKE16562,X-1:CLEAR50
160 X=PEEK(16562)+1:A=X*256:IFA>32767THENA=A-65536
170 DK=NOT(PEEK(16396)=201)
180 IFDKTHENPRINT"DISK";ELSEPRINT"TAPE";
190 PRINT" SYSTEM":PRINT

```



```

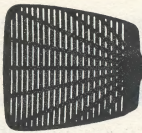
200 PRINT"LOADING STOMP...";CHR$(29);
210 T=0:FORB=0T0491
220 READ: T=T+D: IFD<0 THEN D=ABS(D+1)+X
230 POKEA+B,D
240 NEXTB
250 IFT<>44626 THEN PRINT"SORRY, BUT THERE IS AN ERROR IN THE DATA
    STATEMENTS IN":PRINT"LINE 1000-1300.":PRINT:END
260 IFDK THEN DEFUSR=AELSE POKE16526,PEEK(16561):POKE16527,PEEK(165
    62)
270 U=USR(0):CLEAR50
280 PRINT"STOMP IS UP AND RUNNING.":PRINT
290 PRINT"TO SHUT STOMP OFF, TYPE: ";CHR$(34);"NAME *":CLEAR 50"
    ;CHR$(34):PRINT
1000 DATA 62,195,50,142,65,33,30,-1,34,143,65,33,17,-1,34,177
1010 DATA 64,201,0,0,0,0,0,0,0,0,0,0,0,175,50,29,-1,50,156
1020 DATA 64,43,215,56,40,254,207,32,22,33,143,65,54,74,35,54
1030 DATA 30,33,254,255,34,177,64,62,42,205,42,3,195,25,26,62
1040 DATA 12,50,18,-1,17,244,1,237,83,19,-1,24,30,205,55,35,229
1050 DATA 205,127,10,125,50,18,-1,225,35,205,55,35,229,205,127
1060 DATA 10,34,19,-1,225,126,254,44,32,14,35,126,203,175,254
1070 DATA 80,32,6,62,1,50,156,64,35,34,27,-1,42,164,64,229,175
1080 DATA 79,50,25,-1,221,33,0,0,253,33,0,0,229,22,0,225,229
1090 DATA 126,35,182,40,81,35,35,35,175,71,50,26,-1,126,183,40
1100 DATA 82,254,34,40,47,58,26,-1,183,32,17,126,254,32,40,33
1110 DATA 254,10,40,29,254,58,40,37,254,147,40,55,126,254,97
1120 DATA 56,9,254,123,48,5,214,32,24,1,126,4,169,12,12,95,253
1130 DATA 25,35,24,201,58,26,-1,47,50,26,-1,24,236,35,126,43
1140 DATA 254,147,40,15,24,227,205,114,-2,225,225,42,27,-1,201
1150 DATA 24,155,24,139,120,183,40,67,88,22,0,221,25,225,229
1160 DATA 35,35,94,35,86,237,83,23,-1,33,25,-1,126,183,32,4,237
1170 DATA 83,21,-1,213,22,0,123,169,12,12,95,253,25,209,122,22
1180 DATA 0,169,12,12,95,253,25,52,58,18,-1,190,40,21,42,19,-1
1190 DATA 221,229,209,183,237,82,225,229,56,8,225,94,35,86,235
1200 DATA 229,24,173,205,114,-2,58,156,64,183,32,27,33,29,-1
1210 DATA 52,126,254,15,32,18,175,119,33,64,56,203,70,32,252
1220 DATA 203,86,194,25,26,203,70,40,247,225,94,35,86,235,24
1230 DATA 132,58,25,-1,183,200,42,21,-1,6,7,205,205,-2,62,45
1240 DATA 205,42,3,42,23,-1,6,7,205,205,-2,6,6,205,228,-2,253
1250 DATA 229,225,17,164,2,183,237,82,48,251,25,17,26,0,62,64
1260 DATA 60,183,237,82,48,250,25,205,42,3,125,198,65,205,42
1270 DATA 3,6,8,205,228,-2,221,229,225,6,6,205,205,-2,62,13,205
1280 DATA 42,3,58,64,56,203,87,194,25,26,201,197,205,154,10,175
1290 DATA 205,52,16,182,205,217,15,193,126,183,40,6,205,42,3
1300 DATA 35,16,246,62,32,205,42,3,16,249,201

```



STOMP

STOMP



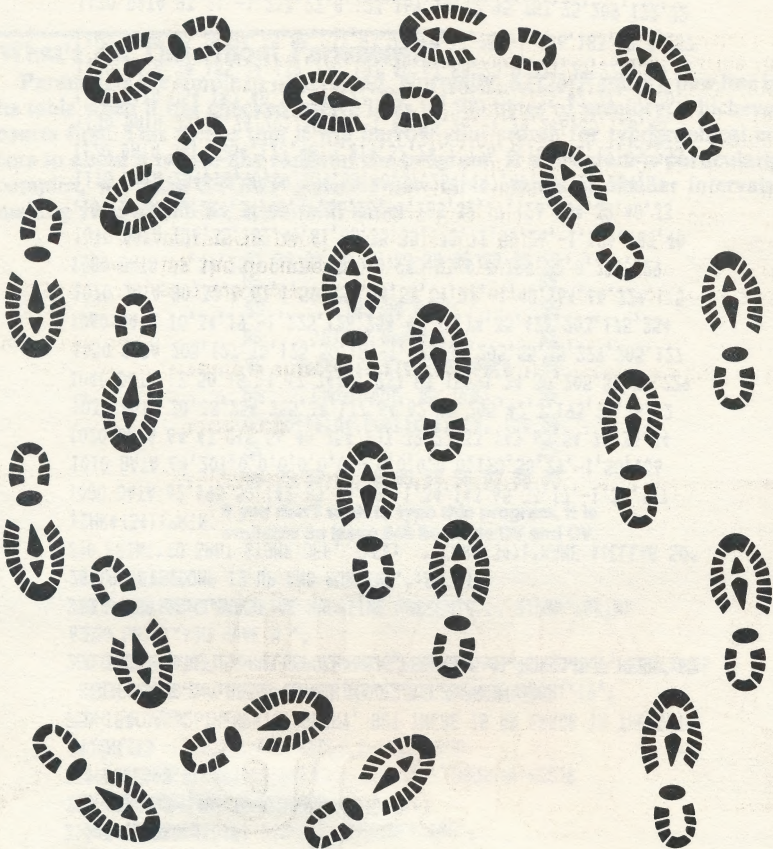
SWAT
TABLE

For TRS-80® STOMP GENERATOR

(Modified Parameters: NU = 3, B = 200)

LINES	SWAT CODE	LENGTH
100 - 120	BC	75
130 - 150	NF	124
160 - 180	IP	94
190 - 210	YH	65
220 - 240	LU	47
250 - 270	VV	157
280 - 1000	UT	158
1010 - 1030	Mk	173
1040 - 1060	EV	177

LINES	SWAT CODE	LENGTH
1070 - 1090	EQ	175
1100 - 1120	IK	174
1130 - 1150	HS	173
1160 - 1180	UM	178
1190 - 1210	RS	173
1220 - 1240	PQ	172
1250 - 1270	AY	175
1280 - 1300	WY	159



Seabase Survival

by Michael P. McEwen

Seabase Survival is a simulation game for any TRS-80® Model I or III with 16K RAM.

As your transport sub descends into the briny ocean depths, you break open the seal protecting the documents describing your secret mission. The eerie glow from the instrument panel illuminates the papers. Your goal is to save Atlantis II, a seabase resting on the ocean floor just off the continental shelf. Atlantis II is powered by a nearby "hot spot" beneath the ocean floor. But alas, this geothermal region's temperature is dropping and technicians are feverishly developing a new technology based on ocean pressure differentials and hydraulics. You have been chosen to manage Atlantis II during this critical period. Your primary objective is to get the new technology going within ten weeks, before the old geothermal system finally fails. During this time, you must keep the small city alive with ample supplies of fresh water and algae.

Your most important resource is the hot sea water spewing from the ocean floor. From the colony's main intake, you can divert variable proportions to the three main systems: the electrical generators, the algae farms and desalinization plant (yielding fresh water for drinking). The number of kilowatt-hours of power generated each week depends on the rate of flow (gallons per second or gal/s) and the temperature. The farms, the fresh water plant, the hydrodome protecting Atlantis II and the new power plant construction all require power.

Algae feeds the people of Atlantis II and depends on water flow rate and power allotment. Plankton is harvested to feed the surrounding whale population and is essentially free, except for the power needed to operate the harvesting machines. In return for plankton, the whales keep your water intake free from debris. Thus, keeping the whales happy is an important factor in your weekly water flow value.

The amounts of power and water assigned to desalinization determine the amount of fresh water available for drinking. You must monitor the air's oxygen content, which changes according to algae production and desalinization.

A hazard of locating Atlantis II near a geothermal area is frequent tremors and landslides. The hydrodome is powered to protect Atlantis II from the in-

SEABASE SURVIVAL

SEABASE SURVIVAL

tense ocean pressure and falling boulders. To help with the latter, a sonic zipper guns down large boulders before they hit the hydrodome during a landslide. Your hydrodome can sustain ten boulder hits before it collapses.

You are interrupted by a jolt as the submarine comes to a halt at the Atlantis II airlock. It is week one and time to begin your work.

Seabase Survival primarily uses the arrow keys and space bar except to input the power distribution. You may use the Alpha Joystick in lieu of the keyboard. Move the joystick in the desired arrow direction and press the fire button instead of the space bar. Sound is available by connecting an amplifier to the cassette AUX port.

Strategies

- During a landslide, horizontally direct the “+” to a falling boulder by arrows or joystick. Press the space bar or the fire button momentarily to unleash an invisible zap.
- Your new technology will develop faster in later weeks so don’t be disappointed that progress seems slow at first. You should assign most of your power to the new technology every week.
- Don’t produce more than the maximum algae needed — but never produce less than half this value.
- Stay on good terms with the whales to achieve maximum water intake.
- The range of acceptable desalinization assignments has a sharp cut-off since reserves hold close to the weekly demand of around 30,000 gallons, so watch out.

Variables

A: Dummy variable used to call sound routine.
 B: Number of boulders left in landslide.
 C: Atmosphere oxygen percentage.
 D: Number of boulder hits hydrodome can sustain.
 DS: Dome status percentage.
 EA: Maximum tons of algae required for current week.
 F: Number of zaps available.
 FP: Zap target crosshair position.
 G: Geothermal region temperature.
 J,K,L: Temporary integer values.
 LP(*): Last week’s power assignments.
 MM(*): Machine language sound routine.
 N: New technology completion percentage.
 NP: New change in fresh water reserves.
 P: Power generated during current week.
 P(*): Power assignments for current week.
 PA: Power assigned to grow algae.

PP: Power assigned to harvest plankton.
 PR: Power not assigned to anything.
 R: Whale status.
 S: Temporary input string.
 S1, S2, S3, S4, S5: Graphic pictures.
 SD: Graphic picture of dome.
 SN(*): System names.
 SP: Prompt for user.
 Note: All variables beginning with the letter “S” are string variables.
 T: Current week number.
 W,WT: Temporary water flow values.
 WA: Water output temperature.
 WC: Water flow rate for current week.
 WH: Desalinization water flow.
 WN: Water intake temperature.
 WR: Quantity of fresh water in reserve.
 X,Y: Falling boulder position.
 YA: Tons of algae produced.
 YP: Tons of plankton harvested.
 Z: Keyboard/joystick input value.


```

SS SS SS SS SS SS SS SS SS SS
SS
SS TRS-80 BASIC SS
SS 'Seabase Survival' SS
SS Author: Michael P. McEwen SS
SS Copyright © 1983 SS
SS SoftSide Publications, Inc SS
SS SS
SS SS SS SS SS SS SS SS SS SS

```

**If you don't wish to type this program, it is
available on issue #45 SoftSide DV and CV.**

Jump to initialization.

```
100 GOTO1000
```

Display picture of Atlantis II.

```

110 PRINT@963,STRING$(58,176);:PRINT@968,SD;:FORJ=127047:SET(0,J
):SET(127,J):IFJ>17THENSET(1,J):SET(126,J):IFJ>23THENSET(2,J):SE
T(125,J):IFJ>32THENSET(3,J):SET(4,J):SET(123,J):SET(124,J):IFJ>4
1THENSET(5,J):SET(122,J)
120 NEXTJ:FORJ=970701012STEP2:PRINT@J,CHR$(143+RND(48));:NEXTJ:R
ETURN

```

Create power generation display.

```

130 A=USR(14000)+USR(14004):CLS:OUT236,16:W=WC:PRINTTAB(5);"ATLA
NTIS II GEOTHERMOELECTRIC POWER GENERATION SYSTEM":PRINT@664,"MA
X FLOW AVAILABLE WEEK";T;":":W;"GAL/S":PRINT@769,S1;:PRINT@833,S
2;
140 FORJ=97044:SET(12,J):IFJ>12ANDJ<33THENSET(16,J)
150 NEXTJ:PRINT@264,S5;S3;:FORJ=167020:SET(48,J):NEXTJ:PRINT@283
,S4;
160 PRINT@129,"WATER IN";:PRINT@152,"GENERATOR";:PRINT@217,"TURB
INE";:PRINT@299,"POWER";:PRINT@329,"FILTER";:PRINT@468,"WATER O
UT";:PRINT@782,"GEOTHERMAL";:PRINT@846,"REGION";
170 PRINT@732,"<UP-ARROW> TO INCREASE WATER FLOW";:PRINT@796,"<D
OWN-ARROW> TO DECREASE WATER FLOW";:PRINT@860,"<SPACE> TO SET GEOT
HERMAL WATER FLOW";

```

Compute intake water and geothermal temperatures.

```

180 WN=RND(11)+39:G=2000*COS(.14*(T-1))+RND(100):PRINT@192,STR$(
WN);CHR$(129);"F";:PRINT@911,STR$(G);CHR$(129);"F";:WT=W

```

**Determine output temperature and power generated. Adjust
water flow according to player input.**

```

190 A=USR(700):WA=G*(2-EXP(WT/2000))/2:P=(WA-WN)*WT/250:PRINT@64
,WT;"GAL/S";CHR$(30);:PRINT@306,P;"MW";CHR$(30);:PRINT@533,STR$(
WA);CHR$(129);"F ";
200 Z=(255-INT(0))ORPEEK(14400)/8:IFZ=1ANDWT<WTHENWT=WT+5:GOTO17
0ELSEIFZ=2ANDWT>5THENWT=WT-5:GOTO190ELSEIFZ<>16THENSET(13+RND(9)
,32+RND(12)):RESET(13+RND(8),32+RND(12)):GOTO200

```


SEABASE SURVIVAL

Display previous week's power assignments and get distributions for current week.

```
210 W=W-WT:A=USR(10000):CLS:OUT236,16:PRINT"YOU ARE GENERATING";
P;"MEGAWATTS OF POWER THIS WEEK.":PRINT:PRINT"PLEASE DISTRIBUTE
THIS ELECTRICITY AMONG THESE SYSTEMS.":PRINT:PRINTTAB(16);"LAST
WEEK";TAB(30);"THIS WEEK"
220 PRINTTAB(16);"-----";TAB(30);"-----":FORJ=1TO4:LP(J)
=P(J):PRINT:PRINTCHR$(30);USING"%          Z #####";SN(J),LP
(J):NEXTJ
230 PR=P:FORJ=1TO4:PRINT@962," UNALLOCATED POWER.":PR;"MEGAWATT
S";CHR$(30);:PRINT@351+J*128,CHR$(30);:INPUTP(J):PR=PR-P(J):NEXT
J
240 IFPR>=0THEN270
250 A=USR(0):PRINT@962,CHR$(30);"YOU HAVE DISTRIBUTED TOO MUCH P
OWER. HIT <ENTER> TO REENTER.":
260 IFINKEY$(<>)CHR$(13)THEN260ELSE230
270 PRINT@964,"<SPACE> IF DISTRIBUTION IS CORRECT OR <ENTER> TO
REENTER.":
280 S=INKEY$:Z=INP(0):IFS="ANDZ=255THEN280ELSEIF(Z>239ANDZ<255)
ORS=CHR$(13)THENPRINT@960,CHR$(30);:GOTO230ELSEIFS<>"ANDZ<>239
THEN280
```

Print water flow available for farming and desalinization, and get player's assignments to both.

```
290 A=USR(10002):CLS:OUT236,16:PRINT"YOU HAVE";W;"GAL/S OF WATER
NOT USED IN POWER GENERATION.":PRINT:PRINT"HOW MANY GAL/S DO YO
U WANT TO USE FOR FARMING"
300 PRINT@192,CHR$(30);:INPUTWT:IFWT>WTHENA=USR(0):GOTO300ELSEWH
=W-WT:PRINT:PRINT"THAT LEAVES";WH;"GAL/S FOR DESALINATION.":PRIN
T@704,"<SPACE> IF CORRECT OR <ENTER> TO REENTER."
310 S=INKEY$:Z=INP(0):IFS="ANDZ=255THEN310ELSEIF(Z>239ANDZ<255)
ORS=CHR$(13)THENPRINT@704,CHR$(30);:PRINT@320,CHR$(30);:GOTO300E
LSEIFS<>"ANDZ<>239THEN310ELSEA=USR(10004)+USR(10000)
```

Set up sea farming display and determine expected demand.

```
320 CLS:OUT236,16:IFP(1)=0THEN410ELSEPRINTTAB(16);"ATLANTIS II'S
EA FARMING DURING WEEK";T:PRINT:PRINT"PROJECTED YIELDS.":PRINT"A
LGAE":PRINT"PLANKTON":PRINT:PRINT"PERCENTAGE OF ELECTRICAL POWER
USED:"
330 PRINT"(TOTAL FARMING ALLOTMENT =";P(1);"MW)":PRINT"ALGAE":PR
INT"PLANKTON":EA=EA+(YA<EA)*5+RND(5):PA=P(1)/2+1:IFEA<1THENA=1
340 PRINT:PRINT"ATLANTIS II WILL NEED, AT MOST,";EA;"TONS OF ALG
AE THIS WEEK.":PRINT:PRINTTAB(9);"<UP-ARROW> TO INCREASE POWER A
LLOTTED FOR ALGAE":PRINTTAB(9);"<DN-ARROW> TO DECREASE POWER ALL
OTTED FOR ALGAE"
350 PRINTTAB(9);"<SPACE> TO SET FARMING POWER DISTRIBUTION";
```

Compute algae/plankton yields and percentages. Display yields in bar graph form and adjust numbers as dictated by player input. Check for starvation.

```
360 PP=P(1)-PA:YA=(1-EXP(-PA/500))*WT:YP=PP/50:K=PA/P(1)*100:L=1
00-K:IFYA>90THENYA=90
```

```

370 A=USR(750):PRINT@521,K;"% ":PRINT@585,L;"% ":PRINT@201,S
  TRING$(YA/2+.6,143);YA;"TON";CHR$(30);CHR$(83+51*(YA=1));
380 PRINT@265,STRING$(YP/2+.6,143);YP;"TON";CHR$(30);CHR$(83+51*
  (YP=1));
390 Z=(255-INP(0))ORPEEK(14400)/8:IFZ=1ANDPA<P(1)-4THENPA=PA+5:G
  OTD360ELSEIFZ=2ANDPA>5THENPA=PA-5:GOTO360ELSEIFZ<>16THEN390
400 CLS:IFYA>EA/2THEN420
410 PRINT"YOU STARVED ATLANTIS II!":GOTO630

```

Begin status report by computing amount of water desalinated and the change in the water reserves. Calculate and print oxygen content of atmosphere.

```

420 PRINT"STATUS OF ATLANTIS II AT THE END OF WEEK";T:A=USR(1025
  0)+USR(10248):PRINT:NP=P(2)/.8:IFWH>NPTHENWH=NP
430 NP=(WH-56+RND(9))*604.8:PRINT"PURE WATER PRODUCED THIS WEEK:
  ";WH*604.8;"THOUSAND GALLONS":IFWR+NP>30000THENNP=30000-WR:WR=30
  000ELSEWR=WR+NP
440 PRINT"NET CHANGE IN WATER RESERVES:";NP;"THOUSAND GALLONS":I
  FWR<1THENPRINT:PRINT"ATLANTIS II RAN OUT OF WATER!":GOTO630
450 PRINT"TOTAL PURE WATER RESERVES:";WR;"THOUSAND GALLONS":PRIN
  T:C=C+WH/9-YA/6:IFC>95THENC=95ELSEIFC<10THENPRINT"ATLANTIS II SU
  FFOCATED!":GOTO630
460 PRINT"ATMOSPHERE:";C;"PERCENT OXYGEN":PRINT

```

Handle new technology development and whale situation.

```

470 N=N-(P(4)>10)*15*SIN(.014*N)+P(4)/100:IFN>=100THEN620ELSEPRI
  NT"NEW TECHNOLOGY IS";N;"PERCENT COMPLETE.":PRINT:IFT=10THENPRIN
  T"ATLANTIS II IS LOST!":GOTO630
480 R=YP+(R<2)-1:IFR<0THENR=0
490 PRINT"THE WHALES ARE ";:IFR<1THENPRINT"ENRAGED!":WC=WC/2ELSE
  IFR<2THENPRINT"DISGRUNTLED.":WC=WC*.8ELSEIFR<3THENPRINT"CONTENT.
  "ELSEPRINT"ELATED!":WC=WC*1.2
500 SP="TO CONTINUE":GOSUB660

```

Check for landslide. If one occurs, determine the number of boulders and available zaps. Display falling boulder and handle player actions.

```

510 IF(R>10RP(3)>200)ANDRND(3)<3THEN570ELSECLS:OUT236,16:PRINT@4
  08,"L A N D S L I D E":FORJ=900TO999:A=USR(J):NEXTJ:GOSUB110:PRI
  NT@408,CHR$(209);:B=RND(T)+5:F=P(3)/2500*DS+RND(5)+1:FP=31
520 X=RND(91)+17:Y=0:B=B-1:A=USR(800)
530 SET(X,Y):SET(X+1,Y):PRINT@730,F;"ZAPS LEFT ";:PRINT@320+FP,"
  +";:Z=(255-INP(0))ORPEEK(14400)/8:IFZ=4ANDFP>3THENPRINT@320+FP,"
  ";:FP=FP-2:GOTO550ELSEIFZ=8ANDFP<60THENPRINT@320+FP," ";:FP=FP+
  2:GOTO550
540 IFZ>=16ANDF>0THENF=F-1:IFY<19ANDY>13ANDABS(2*FP-X)<5THENRESE
  T(X,Y):RESET(X+1,Y):FORJ=1TO20:K=X+RND(3)-2:L=Y+RND(3)-2:SET(K,L
  ):A=USR(1000):RESET(K,L):A=USR(1020):NEXTJ:GOTO560
550 RESET(X,Y):RESET(X+1,Y):Y=Y+1:IFPOINT(X,Y)ORPOINT(X+1,Y)THEN
  FORJ=1TO10:SET(X,Y):SET(X+1,Y):RESET(X,Y):RESET(X+1,Y):A=USR(500
  0):NEXTJ:D=D-1ELSE530
560 IFD>0ANDB>0THEN520

```


SEABASE SURVIVAL

Compute and display hydrodome status.

```
570 CLS:OUT236,16:IFP(3)>200THENDS=D*10ELSEDS=D*P(3)/20
580 PRINT"THE HYDRODOME IS OPERATING AT";DS;"PERCENT CAPACITY.";
PRINT:A=USR(1054)+USR(1030)+USR(1045):IFDS=0THENPRINT"THE DOME C
OLLAPSES!":GOTO630ELSEPRINT"IT CAN SUSTAIN";D;"MORE BOULDER HITS
"
```

Check for excessive water flow and begin new week.

```
590 IFWC>1200THENWC=1200
600 T=T+1:SP="TO BEING WEEK"+STR$(T):GOSUB660
610 GOTO130
```

Handle end of game.

```
620 PRINT"THE NEW TECHNOLOGY IS OPERATIONAL...YOU SAVED ATLANTIS
  II!!":FORJ=300TO350:A=USR(J):NEXTJ:GOTO640
630 A=USR(10)
640 SP="FOR ANOTHER GAME":GOSUB660
650 RUN
```

Subroutine to pause until player hits a key or moves the joystick.

```
660 PRINT@960,"PRESS ANY KEY OR MOVE JOYSTICK ";SP;:S=INKEY$
670 S=INKEY$:IFS=""ANDINP(0)=255THEN670ELSEReturn
```

Initialization: Define variables and sound routine.

```
1000 CLEAR400:RANDOM=DEFINTA-Z:DEFSGN:DEFSTRS=DIJ,K,L,A,B,C,D,
DS,EA,F,FP,G,N,MP,P,PA,PP,PR,R,S,T,WC,W,WT,WA,WH,WN,WR,X,Y,YA,YF
,Z,S1,S2,S3,S4,S5,S6,SP,P(4),LP(4),SN(4),MM(12)
1010 CLS:PRINT@144,"S E A B A S E   S U R V I V A L":PRINT@272,
"COPYRIGHT 1983 MICHAEL P. McEWEN"
1020 S1=CHR$(150)+CHR$(27)+CHR$(152)+CHR$(134)+CHR$(131)+CHR$(27
)+CHR$(148)+CHR$(25)+CHR$(132)+CHR$(25)+CHR$(140)+CHR$(172)+CHR$
(176)+CHR$(26)+CHR$(137)+CHR$(144)+CHR$(26)+CHR$(24)+CHR$(170)
1030 S2=CHR$(181)+CHR$(26)+CHR$(137)+CHR$(148)+CHR$(26)+CHR$(24)
+CHR$(131)+CHR$(131)+STRING$(6,140)+CHR$(131)+CHR$(27)+CHR$(184)
+CHR$(133)+CHR$(27)+CHR$(24)+CHR$(149)
1040 S3=CHR$(174)+CHR$(149)+CHR$(27)+CHR$(24)+CHR$(180)+CHR$(176
)+CHR$(144)+CHR$(26)+CHR$(24)+CHR$(179)+CHR$(179)+CHR$(140)+CHR$
(26)+STRING$(5,24)+CHR$(135)+CHR$(131)+CHR$(151)
1050 S4=STRING$(15,134):S5=CHR$(156)+CHR$(140)+STRING$(4,179)+ST
RING$(7,140)
1060 J=VARPTR(MM(0)):FORK=0TO12:READMM(K):NEXTK:IFPEEK(16396)<>2
01THENCMD"T":DEFUSR=J:GOTO1080
1070 K=J/256:L=J-K#256:POKE16526,L:POKE16527,K
1080 WC=500:WR=10000:SN(1)="FARMING":SN(2)="DESALINATION":SN(3)=
"HYDRODOME":SN(4)="NEW TECHNOLOGY":C=20:T=1:R=2:D=10:EA=40
1090 FORJ=1TO72:READK:IFK>0THENS=D+SD+CHR$(K+126)ELSEIFK=-1THENS
D=SD+CHR$(27)+CHR$(24)ELSEIFK=-2THENS=D+SD+CHR$(26)+CHR$(24)
1100 NEXTJ:GOSUB110
1110 FORJ=2800TO2700STEP-2:A=USR(J):NEXTJ:GOTO130
1120 DATA32717,15882,3585,17919,-6609,3,-45,10253,4100,6390,9714
,-3808,201
```

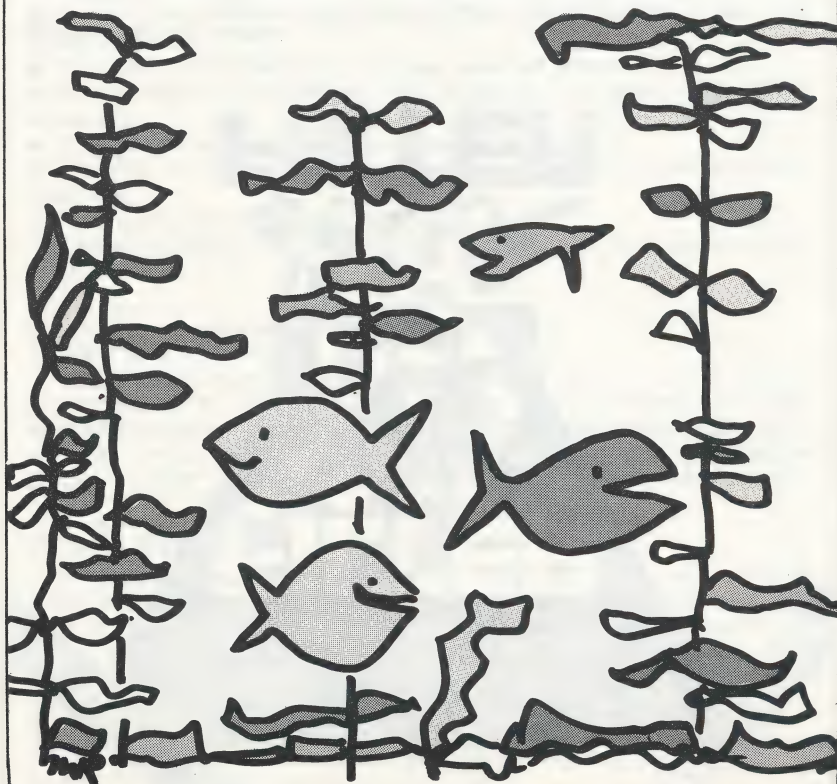
1130 DATA 54,-1,32,7,-1,32,30,1,-1,56,6,1,-1,48,12,3,1,-1,48,28,
12,3,1,-1,32,48,28,12,12,3,3,3,3,3,3,3,3,3,3,12,12,44,48
16,-2,2,3,12,44,48,-2,2,3,12,48,-2,2,9,52,-2,2,45,15,-2,11,16,-
2,57

STOMP TABLE

For TRS-80® SEABASE SURVIVAL

LINES	STOMP CODE	LENGTH
100 - 160	LP	581
170 - 210	QT	640
220 - 290	NO	600
300 - 330	BJ	520
340 - 390	VU	532
400 - 460	IQ	516

LINES	STOMP CODE	LENGTH
470 - 530	BQ	605
540 - 620	DH	583
630 - 1030	BD	571
1040 - 1120	OE	559
1130 - 1130	XP	187



SEABASE SURVIVAL

FILTER

DV BONUS



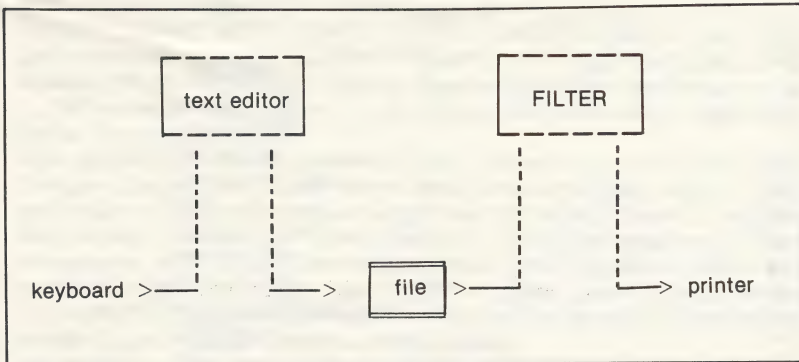
by Michael J. Sorens

Filter is a driver program for the TRS-80® Model I or Model III computer with a Centronics™ 737/739 (Line Printer IV), or an Epson MX™ series printer, with Grafrax™. It may also work with other printers, but some printer-specific commands may not work.

Filter takes a text file created with any word processor, such as *Microtext* 1.2 or 2.0, and “filters” out certain characters to generate stylish output on the printer. Some of the possible features, depending on the printer, are: italics, double width, underlining, condensed width, emphasized type, overprinting, proportional type, subscripting, superscripting, indentation, forward and reverse line feeds, page formatting, header or footer printing with page numbering, and multiple line spacing. In addition, you may add text interactively to the document while it is printing, or display all or part of the unformatted document on the screen with or without line numbers attached. Pertinent statistics on the document, such as average line length and total document size, are easily available. You can modify all the control capabilities, and all require only one or two keystrokes.

Filter Flow Chart

To use *Filter*, enter a document through the keyboard into a computer file using a text editor. *Filter* processes the special control characters, and produces polished output.



The complete sequence of operations for preparing a document under TRSDOS is:

- Prepare a text file with a standard editor, inserting control characters and format commands as described later.
- Store the file on a disk in ASCII (character) format.
- From the operating system level (DOS READY) type BASIC.
- In response to Memory size? press ENTER, and to Number of files? type one, or ENTER.
- Type RUN "FILTER/BAS" to begin using the program.
- Set the control features if you want to bypass the defaults (FEATURES command).
- Load a text file into memory (LOAD command).
- Examine the file or file statistics if desired (DISPLAY or STATISTICS command).
- Print the document (PRINT command).
- Repeat any steps above until finished, then conclude the program with the END command.

Using Filter Features

Format specifications and printing functions are the two types of features you will use in tailoring a document. Printing functions require one or two embedded characters to activate a printer-unique function such as underlining or superscripting. In contrast to these "hard" features, format specifications control the "soft" features such as page margins, indentation and blank lines.

Printing Functions

To enable or disable some inherent feature of the printer, such as double-width printing, type a control character in the text at the desired point. A control character is just some character on the keyboard that is designated by the operator to control a particular function. Each function must have a unique control character. Generally, you should use characters like "#", "\$" or "&", that are not needed in the actual text. If you need a control character as a text character, temporarily bypass the special function, as explained below. A set of default control characters is built into *Filter*.

FILTER

DV BONUS

The first occurrence of the control character enables the function; the second insertion disables it. For example, if the character “/” activates the italics mode, place it just before the word or phrase to be italicized, and deactivate it by placing the slash at the end.

/The slash toggles the italics mode/ on and off.

Notice how the control character brackets the words. Most control characters are independent of each other and can be intermixed.

If the italics code is “/” and boldface is “+” you could do this:

You can mix +**boldface**+ and */italics/* freely within the same sentence — even combinations such as +***bold italics***+ are possible.

Unfortunately, each printer has certain combinations it will refuse to interpret as you wish. All these restrictions are buried in the appropriate printer manual. For example, the Centronics printers won't change from proportional or condensed text back to normal *within* one line, but only at the *beginning* of a line. However, you can change from proportional to condensed, or vice versa, anywhere in a line. The Epson printers can't handle subscripts and superscripts simultaneously; this produces normal characters. Similarly, condensed letters are not possible in the emphasized mode.

Note to Centronics users: One printing feature requires a two-character code: the type set specification (normal, proportional, or condensed type). The Centronics printers can use only one of these printing types at a time so the two-character code ensures only one is active. Each type set is identified with a number: 1 = normal, 2 = proportional, and 3 = condensed. To select a type set, then, requires the type set control character followed by the number of the type set.

A *Filter* document, before printing, has many control characters contained within the text. Once you issue the *PRINT* command, it filters each line, converting each control character into a non-printable command for the printer. In the process, line lengths will decrease; this is important to remember when lining up columns or tables. Also, take care when changing character widths and fonts since in *some* fonts the space character also changes size. Thus, if the left margin is fifteen in *normal width* mode, and you change to condensed, the margin shifts to fifteen *condensed width* spaces.

Changing the font at the beginning of a line may not do what you expect due to the way *Filter* prints a line. First, it puts sufficient blanks into the current character set to get to the appropriate margin. Then it prints the line — and any control characters. So, if the printer is in condensed width and the next line begins with a return to normal width code, the margin for the line still has condensed rather than normal spaces, since the new font isn't activated until after the margin is spaced. Subsequent lines, however, will have normal width spaces used for the margin.

Ten functions are available on the Epson, and seven on the Centronics printers. Condensed and proportional type are controlled by the same code on the Centronics. Listed below are the printing functions along with the default control characters for each printer. A given function is used on a particular printer only if a character is specified for it here:

Function	Epson	Centronics
Condensed	@	\$
Double Strike	+	
Double Width	#	#
Emphasized	\$	
Function Escape	)	)
Italics	!	
Keyboard Input	=	=
Proportional		\$
Subscripting	%	%
Superscripting	*	*
Underlining	/	/

The Keyboard Input function pauses during printing to ask for input. Substitute the input text into the document in place of the control character. When a keyboard input character is encountered, the screen displays that line of text, with a graphic block on top of this character, indicating where to insert the new text. The prompt "TEXT:" appears below the line and the system waits for you to enter the new text. Type only the additional text to replace the control character. A good use for this is to personalize a form letter. You can say, in the middle of the letter, "Thank you, Mr. =, for your patronage." When the printer stops, simply insert the correct name.

The Function Escape cancels any control character, so you can use it as a normal text character. You can use the function escape on itself to print the character rather than to escape some other function.

If you have two keyboard input characters on the same line, the screen display shows a different graphic character for each. This removes any ambiguity.

Format Specifications

Filter formats documents with margins, indents, paginations, and so on. Each format specification must be on a line by itself using the following syntax:

fcc fcode (=) value

Fcc is the format control character, fcode is the two-letter format opcode and value is an operand for the format function (although not all format functions require operands). The equals sign is optional; a space character works too. Sometimes using a space is more natural than an equals sign. The fcc used in this discussion is a period. *Scriptsit*TM users may prefer the greater than symbol ">", since some *Scriptsit* commands are valid *Filter* commands and vice versa (provided the equals sign is used). The *Filter* command ".IN +10" or ".IN = +10" indents ten spaces.

Each format function becomes active at the point in the text where it appears. Thus, if 50 lines have been printed and an embedded command changes the page length to 45, it immediately jumps to the next page, and prints only 45 lines. Finally, note that right justification is not provided; each line is simply truncated at the closest word boundary inside the current right margin.

DV BONUS

FILTER

DV BONUS

Description of Format Codes

- **BP** — Bottom of Page — the printer immediately goes to the bottom of the page, prints a footer, advances to the next page, prints a header, then continues printing.
- **ET** — End of Text marker — this *must* be the last line of each file.
- **FT title** — *Footing Title requires the string title*; if not specified then no footing is printed. The footing is printed in the standard (default) character set.
- **HT title** — *Heading Title requires the string title*; if not specified then no heading is printed. The heading is printed in the standard (default) character set. If this format line is the first line in a file, the header is printed on the first page of output; if entered elsewhere, it is printed on the following page.
- **IG char** — This causes the program to **Ignore** the control character *char* on the next line only.
- **IN (sign) value** — **I**ndents text by adjusting the left margin. If the sign, + or - is present, the left margin moves relative to the current value. Not using the equal sign sets the left margin to the given value, which is equivalent to the **.LM** format line.
- **LL (sign) value** — **S**ets **L**ine **L**ength to *value* and changes the right margin accordingly. If the sign, + or - is present, the line length changes by the given *value*. Default = 80, Max = 132.
- **LM value** — **S**ets the **L**eft **M**argin of the page to *value*. Default = 16, Max = 132.
- **LS value** — **S**ets the **L**ine **S**pacing to *value*. Single spaced is one, two is double spaced, etc. Default = 1, Max = 60.
- **PA** — Causes a **P**Ause. The program waits until you press <ENTER>.
- **PL value** — **S**ets the **P**age **L**ength (number of lines per page) to *value*. Default = 56, Max = 60.
- **PN value** — sets the current **P**age **N**umber to *value*. To set a page number other than one for the header on the first page, this line must follow immediately the **.HT** statement in the file. Default = one.
- **PR char** — **S**tart or stop **P**rinting sections of a document. If *char* is **N** (no) then printing ceases until a **.PR Y** command. Default = **Y** (yes).
- **RM value** — **S**ets the **R**ight **M**argin of the page to *value*. Default = 96, Max = 132.
- **SP (sign) value** — **P**rints *value* blank lines (**S**Paces). If the *sign* is "-", *value*, it prints reverse line feeds (Centronics only), which overprint the line. The Epson printers use double strike mode to overprint. If the *value* is too large (i.e. the printer would leave the current page), the number is truncated to advance just to the top (or bottom) of the current page, depending on the *sign*.
- **TI (sign) value** — **T**emporary **I**ndent — performs the same as **.IN** but only for the line immediately following.
- **** string** — This is a comment which is neither printed nor examined by the program; *string* may be anything but it must fit on one line.

Page Numbering

A page number may appear in either a header or a footer string. Use the pound symbol "#" to mark the location where the page number should be inserted. The page number that begins at that column is printed to the right,

followed by the remainder of the header or footer. Hence, the page number is left-justified. Several pound symbols may be used, each adjacent to the previous one, to define a field for the page number. As long as there are more pound symbols than digits in the number, the page number will be right justified, with leading pound symbols printed as blanks. No control characters other than the pound symbol will be recognized in a header or footer.

Examples:

Header String	Page Number	Printer Output
The page is # now.	1	The page is 1 now.
The page is ### now.	1	The page is 1 now.
The page is # now.	666	The page is 666 now.
The page is ### now.	666	The page is 666 now.
The page is ##### now.	666	The page is 666 now.

Page Boundaries

A physical page consists of 66 lines subdivided into four zones: header, text, footer and between-page. The header and footer zones each occupy three lines. The header zone consists of a header string, which is printed on exactly one line, followed by two blank lines. The footer zone consists of two blank lines followed by the single-line footer string. If either the header or footer has not been specified, this will just be another blank line. The text zone occupies the number of lines set by a Page Length format line (.PL). The between-page zone comes last, occupying however many lines are left to account for all 66 lines. The maximum text zone size is 60 lines. At this limit the footer of one page will be followed immediately by the header of the next, with zero lines for the between-page zone.



FILTER

FILTER

DV BONUS

Alternate Function Escape

The format control character “.” is considered a regular text character automatically if it is not the first character in a line, so in most cases it is not necessary to use the function escape on it.

However, if you wish to use the format control character as a regular character at the beginning of a line, use the function escape. An alternate method available for deactivating the special functions of control characters — the .IG format line. This causes the printing scanner to ignore every occurrence of the specified control character in the line which immediately follows the format line. Only one .IG line can apply to any given line of text at one time, though.

Commands

The menu displays six command choices. When entering a command, you may type just the first character, or any number of characters of the command word. Hence, you could type the *LOAD* command as L, LO, LOA or LOAD. Similarly, any arguments to a command may be abbreviated. Thus, to display a file with line numbers beginning at line 25, you could type *DISPLAY NUM 25* or just *D N 25* (see the *DISPLAY!* command). Type all commands in uppercase letters.

The description of each command is preceded by the syntax of the command line for that particular command, following this pattern:

keyword (option) (option)

In this syntactical description, words in uppercase indicate words that should be typed while words in lower-case italic indicates that an appropriate item should be substituted. The keyword will always be in uppercase letters, signifying that the word should be typed. The parentheses indicate the word is an option; it may or may not be typed at the operator's discretion.

Example: *PRINT (PAUSE) (line)*

indicates that the word “PRINT” should be typed, optionally followed by the word “PAUSE”, optionally followed by a *line* such as “1” or “25”, or any valid line number. The parentheses serve only to indicate that a word is optional. The parentheses themselves should *not* be typed.

LOAD (filename)

This command loads a disk file into memory, showing a count of lines as the file is read. The filename need not be entered with the command; if it is not present, you will be prompted for it. While the file is loading, the text is examined for some of the items shown under the *STATISTICS* command, e.g. longest and shortest line lengths, average line length, and document length.

Special Keys: The “CLEAR” will return to the menu anytime during the load operation, retaining the partial file in memory.

Examples: *LOAD*
LOAD FILE1

DISPLAY (NUM) (line)

Use this to display a file on the screen. If NUM is specified, a line number will be attached to each line. If a line number is specified, the file will be displayed beginning with that line. If a number is not supplied, the file will be listed from the beginning of the document. Either or both options may be used, but if both are used they must appear in the order shown.

Special Keys: The "CLEAR" key will stop the listing. The spacebar will pause the listing. Hitting the spacebar again will resume the listing.

Examples:

```
DISPLAY
DISPLAY NUM 1
DISPLAY 53
DISPLAY NUM
```

FEATURES

Asks you to select a set of control characters. A different character must be selected for each feature that you wish to use. If the features are to be changed, they should be modified *before* loading a file. This is not critical, though it will allow the program to recognize format lines accurately, and not include them in the program statistics available from the *STATISTICS* command. The first item is the format control character, used to initiate a format line. The default is a period. Next is the parameter selection method, with two choices provided: 1) specified in file and 2) enter from keyboard.

Specified in File

This parameter refers to the use of the printing function characters for underlining, subscripting, etc. All but the simplest documents use this method. Each function is listed on the screen. Enter the character to be used for the operation, or press <ENTER> to ignore the operation. After entering all items, the screen might appear as shown below (underlined symbols have been typed by the operator). This is the Centronics version; the Epson version differs only in the list of features. The defaults for both printers are listed above.

ENTER FORMAT CHARACTER: .

PARAMETER SELECTION:

- 1) Specified in file
- 2) Enter from keyboard

ENTER CONTROL CHARACTERS FOR:

Typeface	:	\$	Double Width	:	#
Underline	:	<u>/</u>	Subscript	:	%
Superscript	:	*	Keyboard Input	:	≡
Function Escape	:	)			

>>>><ENTER> TO CONTINUE <<<<

FILTER

DV BONUS

— FILTER —

DV BONUS

Enter From Keyboard

To produce a document that is entirely in one style — italics and double width, for instance — you need not insert the control characters in the text itself. You can do this within the *FEATURES* command. In fact, the text is not examined for any function control characters, though it is examined for format lines. As in the previous section, each feature for the appropriate printer will be listed on the screen. However, instead of entering a control character, you type “Y” (yes) or “N” (no) to signify whether the function should be activated or not. Anything other than a “Y” is interpreted as “N”. Here is an example for the Centronics version:

```

ENTER FORMAT CHARACTER:  _  PARAMETER SELECTION:
                               1) Specified in file
                               ■ 2) Enter from keyboard

TYPEFACE:  1) Normal (10 CPI)
            2) Proportional
            ■ 3) Condensed (16 CPI)

Double Width : N      Underline      : N
Subscript    : N      Superscript    : N

>>>> < ENTER > TO CONTINUE <<<<
  
```

The Epson version, again, differs only in the selection of features. Note that here, in contrast with the “Specified in file” section, the Keyboard Input and Function Escape do not appear. It is meaningless to answer “yes” or “no” to these two functions.

Special Keys: The <ENTER> key will skip to the next entry. The <CLEAR> key restarts the *FEATURES* command.

STATISTICS

For the file currently in memory, these statistics are shown: file name, number of lines in the file, longest and shortest line lengths, average line length, and number of characters in the document. Additionally, all parameters selected with the *FEATURES* commands, or the defaults, will be displayed. The total number of lines does not include format lines, empty lines, or lines which only specify a type set (Centronics version). Similarly, the same types of lines are not included in shortest, longest or average line length determinations.

The screen will show the chosen control characters if the *FEATURES* option “Specified in file” was selected (Epson example):

File: SAMPLE

Longest line: 86 Shortest line: 12 Average: 57.2

T O T A L S: 212 lines 12126 chars

Format line initiator - .

Italics	: !	Double Width	: #
Condensed Width	: @	Emphasized	: \$
Double Strike	: +	Underline	: /
Subscript	: %	Superscript	: *
Keyboard input	: =	Function escape	:)

If "Enter from keyboard" was selected in *FEATURES*, however, then the "yes" or "no" status of each printing function will be indicated:

File: SAMPLE

Longest line: 86 Shortest line: 12 Average: 57.2

T O T A L S: 212 lines 12126 chars

Format line initiator - .

Italics : YES Double Width : NO

Condensed Width : YES Emphasized : NO

Double Strike : NO Underline : NO

Subscript : NO Superscript : NO

PRINT (PAUSE) (line)

When outputting a document in memory to the printer, this command checks if the printer is ready (Centronics version). If not ready, you will be asked to prepare it (turn the switch on or place it on line). The file is then processed and printed. Any keyboard input characters encountered will stop printing temporarily while awaiting the supplemental text to be typed by the operator.

Any errors in control characters or format commands will be indicated on the screen. Once an error is encountered in a given line, the remainder of that line usually will not be processed further, but will be printed as is. Thus, control characters beyond the error will print and will not be activated. This could result in very messy output. Finally, the total number of errors will be displayed after the document has been printed.

The "PAUSE" option causes printing to stop at the beginning of each page, allowing the use of standard typing paper, or other non-continuous forms. Printing resumes when you press the <ENTER> key. You also can effect a PAUSE anytime during printing by pressing the spacebar. To resume printing, press the spacebar once more.

If a line number is provided with the *PRINT* command, printing will commence at that line. Otherwise, printing will begin with the first line of the document. Either or both options may be used, but if both are used they must appear in the order shown.

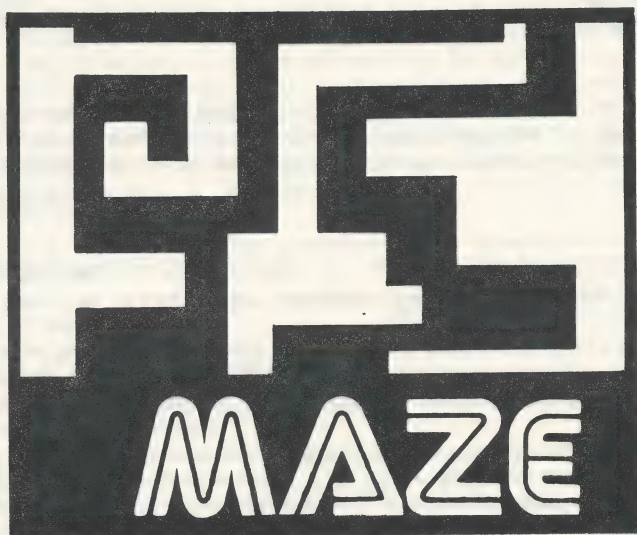
Special Keys: The "CLEAR" key terminates printing. The spacebar pauses or resumes printing.

END (DOS)

This command ends the program. If the "DOS" option is used, the system reverts to the operating system level (DOS READY). Without the "DOS" option, the system reverts to BASIC.

DV BONUS

MAZE



by John F. Gabriel Jr.

Maze is a three-dimensional maze game for a TRS-80® Model I or III with 32K RAM (modifications are included for 16K systems).

With *Maze*, you can create an almost infinite number of different random mazes, and then attempt to solve them — from a mouse's point of view. As the game begins, a three-dimensional diagram of what you can see appears on the screen. The left and right arrow keys change the direction you face 90 degrees left or right, and the down arrow turns you 180 degrees, or reverses your direction. The up arrow moves you forward, and the space bar returns you to the main menu.

Before you enter the maze, a menu presents you with a number of different options. Option zero reenters the maze, while option one restarts you at the beginning of the maze. Neither of these menu options are available until you load or create a maze.

Option two generates a new maze. First, the program prompts you for the width of the maze. Enter a number between five and thirty. If you hit ENTER without answering this question, you return to the main menu. You are then asked for the length of the maze. Enter a number between five and fifteen, or just hit ENTER to change the width of the maze. The computer creates the maze, and returns you to the main menu.

Option three displays the entire maze on the screen, along with your position. You may then press ENTER to return to the menu, or press CLEAR to watch the computer solve the maze. Option four is similar to option three, but prints the maze onto your printer. If you plan to use this option and are printing very wide mazes, you may wish to put your printer into condensed print mode (if possible) before running the *Maze* program.

Option five saves the maze to disk or tape, while option six retrieves a maze from media. As with option two, you may hit ENTER in response to a query to backup to the previous prompt, or to return to the menu.

Option seven prints a page of brief instructions. Option eight is perhaps the most amazing option — it presents a 3D view of the start of the maze, and then solves the maze while presenting a 3D display of each room. The last option, number nine, ends the program.

When you first run the program, only options two, six, seven and nine are available, since no maze is in memory. As soon as you load a maze from disk or tape, or generate a new maze, all of the menu options will be available.

Technical Notes

A two-dimensional array represents the maze internally. Each member of this array is called a room, and is represented by a number. Each number corresponds to a shape which represents the arrangement of walls in that room (see chart A). Note that each room has one to four walls. Four walls in a room represent an empty or unused room. You cannot move into an empty room during the simulation.

The maze is created in two phases. In the first phase, the maze solution is created and stored. In the second phase, the deadends, which shoot off from the solution path, are created. Each of these phases uses the tables to construct the paths that traverse the maze from room to room.

When viewing rooms three-dimensionally, each room within view as you look down the hallway is assigned a number according to Chart B. These numbers represent the position of walls in that room from your viewpoint. If a wall crosses a room, then you cannot see any rooms further down the hallway.

Since a maze and its solution are represented by large arrays, it would take a very long time to store and retrieve mazes. Therefore, the technique of string packing (and unpacking) is used. Instead of saving as many as 950 array values, the values are converted to strings which are concatenated to form larger strings. When a string reaches a predefined length (200 characters), a new string is started. When the program is done, you have as many as five strings which represent all the values of the arrays. We store these strings along with a few other variables. Similarly, when a maze is retrieved, the five strings are loaded. Then, the strings are unpacked and the values are assigned to the arrays.

Note: As listed below, *Maze* runs only on TRS-80s with at least 32K of RAM. The following modifications will make it work on a 16K tape-based system, at the expense of the instruction and disk I/O routines.

```
110 CLEAR 1600
540 IF Q < > 0 THEN PRINT "Y"
560 PRINT "Y"
1220 GOSUB 3000
DELETE 1230-1270
1310 CLS:GOSUB 3000
DELETE 1320-1370
1460 CLS:PRINT "No instructions available.":GOTO 510
DELETE 1470-3050
```


MAZE

Variables

A,B,L,U,V,Z: Temporary values.
A(*,*): Maze layout.
A\$,A1\$: Input variables.
A\$(*): Packed strings for save and load.
A1: ASCII value of A1\$.
B(*,*): Break Main Path table.
C: Number of columns in maze.
C1: Temporary storage of C.
C(*): Column store.
D: Facing direction. 1 = Right, 2 = Up, 3 = Left, 4 = Down (directions when looking at two-dimensional map display).
D(*,*): Main Path Direction table.
E: Temporary direction and flag.
E(*,*): Table for returning room type for proper direction.
F: First room.
G: Second room.
G\$,G1\$,G2\$,G3\$,G4\$,G5\$,G6\$,G7\$,G8\$,G9\$: Various graphic pictures used to create the 3D maze displays.

H,I,J,K: Third, fourth, fifth and sixth room.
LE: Length of input string.
M: Temporary column position.
ML: Maximum length allowed for input string.
N: Temporary row position.
P: Last item index of arrays C(*) and R(*).
Q: Entrance column.
QU\$: Equals CHR\$(34), a quotation mark.
R: Number of rows.
R1: Temporary storage of R.
R(*): Row storage.
R\$(*): Graphic pictures of each room, as shown in table A.
S: Last item index of array S(*).
S(*): Solution directions storage.
T: Used to PEEK keyboard memory.
W: Temporary storage of last direction.
X,Y: Player's column and row position.

Numbers Assigned	Chart A	Chart B
0		(Nothing)
1	 (Right)	 (Wall)
2	 (Top)	 (Dead-End)
3	 (Left)	 (Wall leading right)
4	 (Bottom)	 (Wall leading left)
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

```

SS SS SS SS SS SS SS SS SS SS SS
SS                                     SS
SS      TRS-80 BASIC                  SS
SS      'Maze'                        SS
SS Author: John F. Gabriel Jr.        SS
SS      Copyright © 1983              SS
SS SoftSide Publications, Inc         SS
SS                                     SS
SS SS SS SS SS SS SS SS SS SS SS

```

If you don't wish to type this program, it is available on issue #45 SoftSide DV and CV.

Initialization.

```
100 RANDOM:CLS:PRINTTAB(14);"MAZE":PRINTTAB(5);"BY JOHN F. GABRI
EL JR."
```

```
110 CLEAR2000
```

```
120 DEFINT A-Z:GOSUB5000
```

```
130 GOTO510
```

Subroutine to input a maximum of
(ML) characters into variable A\$.

```
140 LE=0:A$="":PRINTCHR$(143);:A1$=INKEY$
```

```
150 A1$=INKEY$:IFA1$=""THEN150
```

```
160 A1=ASC(A1$):IFA1>96ANDA1<123THEN A1=A1-32
```

```
170 IFA1=32ANDLE<>MLTHENPRINTCHR$(24);CHR$(A1);CHR$(143);:LE=LE
+1:A$=A$+CHR$(A1):GOTO150
```

```
180 IFA1=13THENPRINTCHR$(8);:A=VAL(A$):RETURNELSEIFA1=8ANDLE<>0T
HENPRINTCHR$(8);CHR$(8);CHR$(143);:LE=LE-1:A$=LEFT$(A$,LE):GOTO1
50
```

```
190 IFA1=24ANDLE<>0THENPRINTSTRING$(LE+1,8);:GOTO140ELSEIFA1=1TH
ENPRINT:PRINT"Break in";PEEK(16546)+PEEK(16547)*256:END
```

```
200 GOTO150
```

Subroutine to erase previous input
and get new input from user.

```
210 PRINTSTRING$(LE,8);:GOTO140
```

Subroutine to input a single
character from user.

```
220 ML=1:GOTO140
```

Get maze dimensions.

```
230 CLS:PRINT"Width of maze (5 - 30) ";:ML=2:GOSUB140
```

```
240 IFLE=0THEN500ELSEC1=A:IFC1<50RC1>30THEN60SUB210:GOTO240
```

```
250 PRINT:PRINT:PRINT"Length of maze (5 - 15) ";:GOSUB140
```

```
260 IFLE=0THEN230ELSER1=A:IFR1<50RR1>15THEN60SUB210:GOTO260
```

```
270 C=C1:R=R1:PRINT:PRINT
```

Create main path (solution).

```
280 PRINT"Thinking...":FORB=1TOR:FORA=1TOC:A(A,B)=0:NEXTA:NEXTB:
P=0:Q=RND(C):A(Q,R)=6:A(Q,R+1)=6:S(0)=2:S(1)=2:S=1:M=Q:N=R-1:W=2
:B=1:D=RND(3):GOTO300
```

```
290 D=RND(4)
```

```
300 Z=0:GOSUB1660:IFZ=4THEN280ELSEA(M,N)=D(D,W):S=S+1:S(S)=D:GOS
UB1760:GOSUB1640:W=D
```


MAZE

```

310 IFZ<>5THEN290ELSEA=0:B=0:CLS:PRINT"Getting there...":PRINT@1
28,"Estimated wait time left":U=C$(R-1)
Create dead ends shooting off main
path.
320 A=A+1:IFA>PTHEN370ELSEPRINT@154,U-A;:M=C(A):N=R(A):D=RND(4):
Z=0:GOSUB1660:IFZ>3THEN320
330 A(M,N)=B(D,A(M,N)):GOSUB1640:L=RND(8):W=D
340 L=L-1:IFL<1THEN350ELSED=RND(4):Z=0:GOSUB1660:IFZ>3THEN350ELS
EA(M,N)=D(D,W):GOSUB1760:GOSUB1640:W=D:GOTO340
350 IFW=1THENA(M,N)=12ELSEIFW=2THENA(M,N)=11ELSEIFW=3THENA(M,N)=
14ELSEA(M,N)=13
360 GOTO320
Put player at entrance.
370 Y=R+1
Check if user's position is at en-
trance or exit.
380 IFY>RTHENF=5:G=0:GOSUB1950:GOSUB2050:X=Q:Y=R+1:D=2:GOTO400EL
SEIFY=0THENCLS:PRINT"Congratulations, you made it out!":GOTO510
390 GOSUB1850
Get keyboard input (arrow key or
space bar).
400 T=PEEK(14400):IFT=128THEN500ELSEIFT=64THENE=1ELSEIFT=32THENE
=3ELSEIFT=16THENE=4ELSEIFT=8THENE=2ELSE400
Perform specified movement.
410 GOSUB1520:GOSUB1610:GOTO380
Display menu and get user selec-
tion.
500 CLS:PRINT
510 PRINT:PRINT"You may do any of the following":PRINT:IFQ<>0TH
ENPRINT"0 - ENTER maze":PRINT"1 - RESTART (go to the entrance of
the maze)"
520 PRINT"2 - GENERATE a new maze"
530 IFQ<>0THENPRINT"3 - REVEAL the maze (2-dimensionally)":PRINT
"4 - PRINT maze on printer":PRINT"5 - SAVE the maze (to tape";
540 IFQ<>0THENPRINT" or disk)"
550 PRINT"6 - LOAD a maze (from tape";
560 PRINT" or disk)":PRINT"7 - Print Instructions"
570 IFQ<>0THENPRINT"8 - Show SOLUTION in 3-D"
580 PRINT"9 - EXIT program"
590 PRINT:PRINT"Your choice ?";:GOSUB220
600 IFA>9OR(A=0ANDL<>0ANDM<>0)OR((Q=0)AND(A<>2ANDM<>6ANDL<>7
ANDM<>9))THENGOSUB210:GOTO600
610 ONA+16GOTO380,370,230,1000,1080,1150,1310,1460,1780,1490
Reveal maze in 2D.
1000 CLS:FORB=1TOR:FORA=1TOC
1010 PRINT$(A(A,B));
1020 NEXTA:PRINT:NEXTB
1030 IFX>0ANDX<=CANDY>0AND(Y<=R+1ANDY<>16)THENA=X#4-3:B=Y#3-2ELS
EA=-1
1040 PRINT@960,"Press <CLEAR> for solution or <ENTER> to continu
e";

```

```

1050 IFA<>-1THENSET(A,B):SET(A+1,B)
1060 T=PEEK(14400):IFA<>-1THENRESET(A,B):RESET(A+1,B)
1070 IFT=1THENS00ELSEIFT=2THENG0SUB1820:G0T01030ELSE1050
Print maze on printer.
1080 CLS:PRINT:PRINT"Do you want to print the maze? (Y/N) ";
1090 ML=3:G0SUB140:IFLEFT$(A$,1)<>"Y"THENS00
1100 FORB=1TOR:FORZ=0TO2
1110 LPRINTLEFT$(P$(A(1,B),Z),1);:FORA=1TOC
1120 LPRINTP$(A(A,B),Z);:NEXTA
1130 LPRINTRIGHT$(P$(A(C,B),Z),1):NEXTZ,B
1140 CLS:PRINT"Printing complete.":G0T0510
Save maze to disk or tape.
1150 CLS:PRINT"Preparing data...":PRINT
1160 FORZ=1TO5:A$(Z)="" :NEXTZ:L=0:Z=1
1170 FORB=1TOR:FORA=1TOC:L=L+1:IFL>200THENL=1:Z=Z+1
1180 A$(Z)=A$(Z)+CHR$(64+A(A,B)):NEXTA,B
1190 L=0:Z=Z+1
1200 FORA=0TOS:L=L+1:IFL>200THENL=1:Z=Z+1
1210 A$(Z)=A$(Z)+CHR$(64+S(A)):NEXTA:CLS
1220 CLS:PRINT"SAVE: ";:G0SUB2970:IFA$=""THENS00
1230 IFA$="C"THEN1280
1240 OPEN"D",1,A$
1250 PRINT#1,Q;"", "C";", "R";", "Z";", "X";", "Y";", "S";", "D
1260 FORA=1TOZ:PRINT#1,A$(A):A$(A)="" :NEXTA
1270 CLOSE1:G0T01300
1280 PRINT#-1,Q,C,R,Z,X,Y,S,D
1290 FORA=1TOZ:PRINT#-1,A$(A):A$(A)="" :NEXTA
1300 CLS:PRINT"Save completed.":G0T0510
Load a maze from disk or tape.
1310 CLS:PRINT"LOAD: ";:G0SUB2970:IFA$=""THENS00
1320 IFA$="C"THEN1380
1330 ONERRORG0T01340:OPEN"I",1,A$:ONERRORG0T00:G0T01360
1340 RESUME1350
1350 CLS:PRINT"Maze";A;"was not found (or other disk error)":G0T
0510
1360 INPUT#1,Q,C,R,Z,X,Y,S,D
1370 FORA=1TOZ:INPUT#1,A$(A):NEXTA:CLOSE1:G0T01400
1380 INPUT#-1,Q,C,R,Z,X,Y,S,D
1390 FORA=1TOZ:INPUT#-1,A$(A):NEXTA
1400 L=0:Z=1:FORB=1TOR:FORA=1TOC:L=L+1:IFL>200THENL=1:Z=Z+1
1410 A(A,B)=ASC(MID$(A$(Z),L,1))-64:NEXTA:NEXTB
1420 L=0:Z=Z+1:FORA=0TOS:L=L+1:IFL>210THENL=1:Z=Z+1
1430 S(A)=ASC(MID$(A$(Z),L,1))-64:NEXTA
1440 FORA=1TOZ:A$(A)="" :NEXTA
1450 CLS:PRINT"Load completed.":G0T0510
Print instructions.
1460 CLS:PRINTTAB(24);"MAZE":PRINT:PRINT" Maze is an ";QU$;"am
azing";QU$;" 3-D simulation in which your":PRINT"task is to get
from one end of a maze to the other."

```


MAZE

```

1470 PRINT:PRINT" You maneuver through the maze using the arrow
w keys.":PRINT"The left and right arrows change your facing, and
the down":PRINT"arrow does an about face. The up arrow moves y
ou forward."
1480 PRINT:PRINT" To get back to the command list,":PRINT"pres
s the space bar instead of an arrow key.":PRINT:PRINT" Press <
ENTER> to continue ";:ML=0:GOSUB140:GOTO500
Exit routine. Before exiting, double
check that user wants to quit.
1490 CLS:PRINT"Are you sure you wish to exit? (Y/N) ";
1500 ML=3:GOSUB140:IFLEFT$(A$,1)<>"Y"THEN500
1510 PRINT:PRINT:END
Turn or move according to user
command. Return with E = 1 to move
in specified direction, or E = 0 to re-
main in place.
1520 ONEGOTO1530,1550,1570,1590
1530 E=0:D=D-1:IFD=0THEND=4
1540 RETURN
1550 IFF<STHENE=0ELSEE=1
1560 RETURN
1570 E=0:D=D+1:IFD=5THEND=1
1580 RETURN
1590 E=0:D=D+2:IFD>4THEND=D-4
1600 RETURN
If E < > 0, move forward one room.
1610 IFE=0THENRETURN
1620 IFD=1THENX=X+1ELSEIFD=2THENY=Y-1ELSEIFD=3THENX=X-1ELSEY=Y+1
1630 RETURN
Change temporary position in maze
according to direction.
1640 IFD=1THENM=M+1ELSEIFD=2THENM=N-1ELSEIFD=3THENM=M-1ELSEN=N+1
1650 RETURN
If room is empty, then return. Other-
wise, change direction and try
another room. If more than three
rooms are tried, return.
1660 V=V+1:IFV>2THENV=1
1670 OND GOTO1680,1690,1710,1720
1680 IFM+1>CTHEN1730ELSEIFA(M+1,N)<>0THEN1730ELSEReturn
1690 IFN-1<1ANDB=1THENZ=5:RETURN
1700 IFN-1<1THEN1730ELSEIFA(M,N-1)<>0THEN1730ELSEReturn
1710 IFM-1<1THEN1730ELSEIFA(M-1,N)<>0THEN1730ELSEReturn
1720 IFN+1>RTHEN1730ELSEIFA(M,N+1)<>0THEN1730ELSEReturn
1730 Z=Z+1:IFV=1THEND=D+1ELSED=D-1
1740 IFZ>3THENRETURNELSEIFD=5THEND=1ELSEIFD=0THEND=4
1750 GOTO1670
Store room coordinates.
1760 P=P+1:IFP>400THENP=400
1770 C(P)=M:R(P)=N:RETURN
Draw solution in 3D.
1780 U=X:V=Y:Y=R+1:F=5:G=0:GOSUB1950:GOSUB2050:X=G:D=2:E=2:FORZ=
1TO100:NEXTZ

```

```

1790 FORL=0TOS:D=S(L):IFE=DTHENPRINT987,"FORWARD";ELSEIFD=E+1OR
(D=1ANDE=4)THENPRINT9454,"LEFT";ELSEPRINT9502,"RIGHT";
1800 E=D:GOSUB1620:IFL<>STHEN60SUB1850ELSEY=1:GOSUB1850:PRINT960
3,"E X I T";
1810 FORZ=1TO150:NEXTZ,L:Y=V:X=U:FORZ=1TO400:NEXTZ:GOTO500
Display solution on 2D display.
1820 PRINT960,CHR$(30);:U=X:V=Y:Y=R+1:X=Q:E=2
1830 FORL=0TOS:D=S(L):A=X#4-3:B=Y#3-2:FORZ=1TO20:SET(A,B):SET(A+
1,B):RESET(A,B):RESET(A+1,B):NEXTZ
1840 E=D:GOSUB1620:NEXTL:Y=V:X=U:RETURN
Draw 3D view of each room down
hallway. If there is no room to draw
or a wall crosses the screen, then
don't draw any room further down
the hallway.
1850 F=0:G=0:H=0:I=0:J=0:K=0:M=X:N=Y:F=E(A(M,N),D):GOSUB1920:IFZ
=1ORF<5THEN1910
1860 G=E(A(M,N),D):GOSUB1920:IFZ=1ORG<5THEN1910
1870 H=E(A(M,N),D):GOSUB1920:IFZ=1ORH<5THEN1910
1880 I=E(A(M,N),D):GOSUB1920:IFZ=1ORI<5THEN1910
1890 J=E(A(M,N),D):GOSUB1920:IFZ=1ORJ<5THEN1910
1900 K=E(A(M,N),D)
1910 GOSUB1950:GOSUB2050:GOSUB2150:GOSUB2250:GOSUB2350:GOTO2450
Determine if next possible room is
beyond maze limits.
1920 IFD=1THENM=M+1ELSEIFD=2THENN=N-1ELSEIFD=3THENM=M-1ELSEN=N+1
1930 IFM>CORN<1THENZ=1ELSEIFN>RORN<1THENZ=1ELSEZ=0
1940 RETURN
Display 3D view of room 1.
1950 CLS:ONF+1GOTO1960,1970,1980,1990,2000,2010,2020,2030,2040
1960 RETURN
1970 GOSUB2550:GOSUB2670:GOTO2910
1980 GOSUB2610:GOSUB2670:GOSUB2730:GOSUB2790:GOTO2850
1990 GOSUB2610:GOSUB2670:GOSUB2730:GOTO2910
2000 GOSUB2550:GOSUB2670:GOSUB2790:GOTO2850
2010 GOSUB2610:GOSUB2730:GOSUB2790:GOTO2850
2020 GOSUB2610:GOSUB2730:GOSUB2790:GOTO2910
2030 GOSUB2550:GOSUB2730:GOSUB2790:GOTO2850
2040 GOSUB2550:GOSUB2730:GOSUB2790:GOTO2910
Display 3D view of room 2.
2050 ONG+1GOTO2060,2070,2080,2090,2100,2110,2120,2130,2140
2060 RETURN
2070 GOSUB2560:GOSUB2680:GOTO2920
2080 GOSUB2620:GOSUB2680:GOSUB2740:GOSUB2800:GOTO2860
2090 GOSUB2620:GOSUB2680:GOSUB2740:GOTO2920
2100 GOSUB2560:GOSUB2680:GOSUB2800:GOTO2860
2110 GOSUB2620:GOSUB2740:GOSUB2800:GOTO2860
2120 GOSUB2620:GOSUB2740:GOSUB2800:GOTO2920
2130 GOSUB2560:GOSUB2740:GOSUB2800:GOTO2860
2140 GOSUB2560:GOSUB2740:GOSUB2800:GOTO2920
Display 3D view of room 3.
2150 ONH+1GOTO2160,2170,2180,2190,2200,2210,2220,2230,2240

```

MAZE

MAZE

2160 RETURN
 2170 GOSUB2570:GOSUB2690:GOTO2930
 2180 GOSUB2630:GOSUB2690:GOSUB2750:GOSUB2810:GOTO2870
 2190 GOSUB2630:GOSUB2690:GOSUB2750:GOTO2930
 2200 GOSUB2570:GOSUB2690:GOSUB2810:GOTO2870
 2210 GOSUB2630:GOSUB2750:GOSUB2810:GOTO2870
 2220 GOSUB2630:GOSUB2750:GOSUB2810:GOTO2930
 2230 GOSUB2570:GOSUB2750:GOSUB2810:GOTO2870
 2240 GOSUB2570:GOSUB2750:GOSUB2810:GOTO2930
 Display 3D view of room 4.
 2250 ONI+16GOTO2260,2270,2280,2290,2300,2310,2320,2330,2340
 2260 RETURN
 2270 GOSUB2580:GOSUB2700:GOTO2940
 2280 GOSUB2640:GOSUB2700:GOSUB2760:GOSUB2820:GOTO2880
 2290 GOSUB2640:GOSUB2700:GOSUB2760:GOTO2940
 2300 GOSUB2580:GOSUB2700:GOSUB2820:GOTO2880
 2310 GOSUB2640:GOSUB2760:GOSUB2820:GOTO2880
 2320 GOSUB2640:GOSUB2760:GOSUB2820:GOTO2940
 2330 GOSUB2580:GOSUB2760:GOSUB2820:GOTO2880
 2340 GOSUB2580:GOSUB2760:GOSUB2820:GOTO2940
 Display 3D view of room 5.
 2350 ONJ+16GOTO2360,2370,2380,2390,2400,2410,2420,2430,2440
 2360 RETURN
 2370 GOSUB2590:GOSUB2710:GOTO2950
 2380 GOSUB2650:GOSUB2710:GOSUB2770:GOSUB2830:GOTO2890
 2390 GOSUB2650:GOSUB2710:GOSUB2770:GOTO2950
 2400 GOSUB2590:GOSUB2710:GOSUB2830:GOTO2890
 2410 GOSUB2650:GOSUB2770:GOSUB2830:GOTO2890
 2420 GOSUB2650:GOSUB2770:GOSUB2830:GOTO2950
 2430 GOSUB2590:GOSUB2770:GOSUB2830:GOTO2890
 2440 GOSUB2590:GOSUB2770:GOSUB2830:GOTO2950
 Display 3D view of room 6.
 2450 ONK+16GOTO2460,2470,2480,2490,2500,2510,2520,2530,2540
 2460 RETURN
 2470 GOSUB2600:GOSUB2720:GOTO2960
 2480 GOSUB2660:GOSUB2720:GOSUB2780:GOSUB2840:GOTO2900
 2490 GOSUB2660:GOSUB2780:GOSUB2840:GOTO2960
 2500 GOSUB2600:GOSUB2780:GOSUB2840:GOTO2900
 2510 GOSUB2660:GOSUB2780:GOSUB2840:GOTO2900
 2520 GOSUB2660:GOSUB2720:GOSUB2780:GOTO2960
 2530 GOSUB2600:GOSUB2720:GOSUB2840:GOTO2900
 2540 GOSUB2600:POKE15839,188:POKE15840,188:POKE15903,143:POKE15904,143:GOTO2960
 Draw portions of rooms.

Draw left hallway horizontal strips.

2550 PRINT#72,67\$,:PRINT#904,69\$,:RETURN
 2560 PRINT#140,68\$,:PRINT#844,68\$,:RETURN
 2570 PRINT#208,69\$,:PRINT#784,67\$,:RETURN
 2580 PRINT#340,67\$,:PRINT#660,69\$,:RETURN
 2590 PRINT#408,68\$,:PRINT#600,68\$,:RETURN

```

2600 PRINT@476,68$;:PRINT@540,68$;:RETURN
Draw left hallway diagonal strips.
2610 PRINT@8,61$;:PRINT@968,64$;:RETURN
2620 PRINT@76,62$;:PRINT@908,65$;:RETURN
2630 PRINT@144,63$;:PRINT@848,66$;:RETURN
2640 PRINT@276,61$;:PRINT@724,64$;:RETURN
2650 PRINT@344,62$;:PRINT@664,65$;:RETURN
2660 PRINT@412,63$;:PRINT@604,66$;:RETURN
Draw wall across screen.
2670 PRINT@75,STRING$(42,131);:PRINT@907,STRING$(42,176);:RETURN
2680 PRINT@143,STRING$(34,140);:PRINT@847,STRING$(34,140);:RETUR
N
2690 PRINT@211,STRING$(26,176);:PRINT@787,STRING$(26,131);:RETUR
N
2700 PRINT@343,STRING$(18,131);:PRINT@663,STRING$(18,176);:RETUR
N
2710 PRINT@411,STRING$(10,140);:PRINT@603,STRING$(10,140);:RETUR
N
2720 POKE15839,140:POKE15840,140:POKE15903,140:POKE15904,140:RET
URN
Draw left hallway vertical lines.
2730 PRINT@75,MID$(6$,3,40);:RETURN
2740 PRINT@143,CHR$(188);LEFT$(6$,32);CHR$(143);:RETURN
2750 PRINT@211,CHR$(176);LEFT$(6$,26);CHR$(131);:RETURN
2760 PRINT@343,MID$(6$,3,16);:RETURN
2770 PRINT@411,CHR$(188);LEFT$(6$,8);CHR$(143);:RETURN
2780 POKE15839,176:POKE15903,131:RETURN
Draw right hallway vertical lines.
2790 PRINT@116,MID$(6$,3,40);:RETURN
2800 PRINT@176,CHR$(188);LEFT$(6$,32);CHR$(143);:RETURN
2810 PRINT@236,CHR$(176);LEFT$(6$,26);CHR$(131);:RETURN
2820 PRINT@360,MID$(6$,3,16);:RETURN
2830 PRINT@420,CHR$(188);LEFT$(6$,8);CHR$(143);:RETURN
2840 POKE15840,176:POKE15904,131:RETURN
Draw right hallway diagonal strips.
2850 PRINT@53,64$;:PRINT@1013,61$;:RETURN
2860 PRINT@177,66$;:PRINT@881,63$;:RETURN
2870 PRINT@237,65$;:PRINT@813,62$;:RETURN
2880 PRINT@297,64$;:PRINT@745,61$;:RETURN
2890 PRINT@421,66$;:PRINT@613,63$;:RETURN
2900 PRINT@481,65$;:PRINT@545,62$;:RETURN
Draw right hallway horizontal strips.
2910 PRINT@117,67$;:PRINT@949,69$;:RETURN
2920 PRINT@177,68$;:PRINT@881,68$;:RETURN
2930 PRINT@237,69$;:PRINT@813,67$;:RETURN
2940 PRINT@361,67$;:PRINT@681,69$;:RETURN
2950 PRINT@421,68$;:PRINT@613,68$;:RETURN
2960 PRINT@481,68$;:PRINT@545,68$;:RETURN

```


MAZE

Subroutine to determine if a save or load is to use disk or tape.
If disk, get maze number. If tape, wait until cassette is ready.

```
2970 PRINTQU$;"D";QU$;" for disk or ";QU$;"C";QU$;" for cassette
- ";:GOSUB220
2980 IFLE=0THENRETURNELSEIFA$="D"THEN3020ELSEIFA$="C"THEN3000
2990 GOSUB210:GOTO2980
3000 PRINT:PRINT:PRINT"Hit <ENTER> when cassette is ready ";
3010 ML=0:GOSUB140:A$="C":RETURN
3020 PRINT:PRINT:PRINT"Enter maze number (1-9999) ";:ML=4:GOSUB1
40
3030 IFLE=0THENRETURNELSEIFA<1000>9999THEN60SUB210:GOTO3030
3040 A$="MAZE"+MID$(STR$(A),2)+"/DAT"
3050 RETURN
```

Subroutine to dimension and initialize variables and arrays.

```
5000 DIMA,A$,A1,B,C,D,E,F,G,6$,61$,62$,63$,64$,65$,66$,67$,68$,6
9$,H,I,J,K,L,LE,M,ML,N,P,Q,QU$,R,S,T,U,V,W,X,Y,Z,ZZ
5010 DIMA(30,16),B(4,10),C(400),R(400),D(4,4),E(14,4),S(400),A$(
5),P$(14,2),R$(14)
5020 6$=CHR$(26)+CHR$(24)+CHR$(191)
5030 FORJ=1TO4:6$=6$+6$:NEXTJ
5040 61$=CHR$(131)+CHR$(140)+CHR$(176)
5050 62$=CHR$(140)+CHR$(176)+CHR$(26)+CHR$(131)
5060 63$=CHR$(176)+CHR$(26)+CHR$(131)+CHR$(140)
5070 64$=CHR$(176)+CHR$(140)+CHR$(131)
5080 65$=CHR$(140)+CHR$(131)+CHR$(27)+CHR$(176)
5090 66$=CHR$(131)+CHR$(27)+CHR$(176)+CHR$(140)
5100 67$=STRING$(3,131):68$=STRING$(3,140):69$=STRING$(3,176)
5110 R$(0)=CHR$(183)+CHR$(187):R$(1)=" "+CHR$(170):R$(2)=STRING$(
2,131):R$(3)=CHR$(149)+" ":R$(4)=STRING$(2,176)
5120 R$(5)=CHR$(147)+CHR$(171):R$(6)=CHR$(149)+CHR$(170):R$(7)=C
HR$(177)+CHR$(186):R$(8)=CHR$(151)+CHR$(163)
5130 R$(9)=STRING$(2,179):R$(10)=CHR$(181)+CHR$(178):R$(11)=CHR$(
151)+CHR$(171):R$(12)=CHR$(179)+CHR$(187)
5140 R$(13)=CHR$(181)+CHR$(186):R$(14)=CHR$(183)+CHR$(179)
5150 QU$=CHR$(34)
5160 FORZ=0TO14:READP$(Z,0),P$(Z,1),P$(Z,2):NEXTZ
5170 DATA ###,###,###," I"," I"," I",IIII," "," "
,"I ","I ","I "," "," ",IIII,IIII," I"," I",I I
,I I,I I
5180 DATA " I"," I",IIII,IIII,"I ","I ",IIII," ",IIII
,"I ","I ",IIII,IIII,"I I","I I",IIII," I",IIII,I I,I
I,IIII,IIII,"I ",IIII
5190 QU$=CHR$(34)
5200 FORA=5TO10:FORB=1TO4:READB(B,A):NEXTB,A
5210 DATA 2,1,5,5,3,6,1,6,4,7,7,1,8,3,2,2,9,4,9,2,10,10,4,3
5220 FORA=1TO4:FORB=1TO4:READD(B,A):NEXTB,A
5230 DATA 9,7,0,5,8,6,5,0,0,10,9,8,10,0,7,6
```

5240 FORA=1T04:FORB=0T014:READE(B,A):NEXTB,A

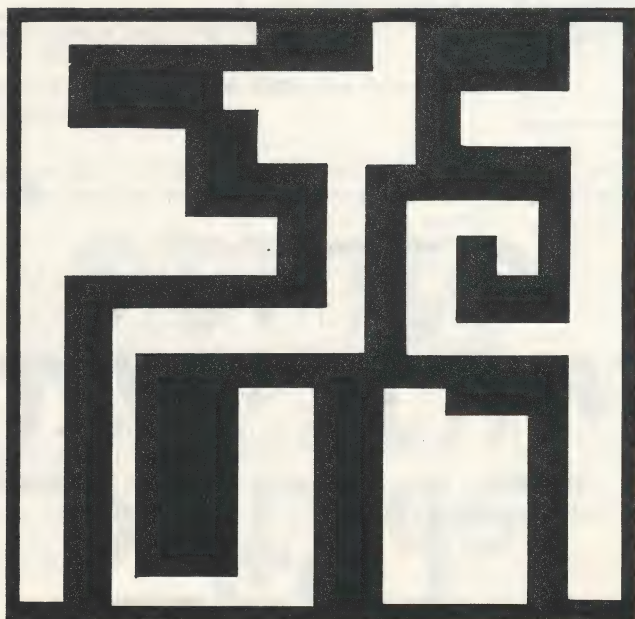
5250 DATA 0,1,6,8,7,3,1,4,6,5,7,3,2,4,5,0,7,1,6,8,4,5,7,3,1,6,2,
4,5,3,0,8,7,1,6,7,1,6,4,5,3,4,5,3,2,0,6,8,7,1,6,5,3,7,1,4,5,3,2,
4

5260 RETURN



For TRS-80® MAZE

LINES	STOMP CODE	LENGTH	LINES	STOMP CODE	LENGTH
100 - 210	OS	371	2000 - 2110	SK	280
220 - 320	VN	513	2120 - 2230	ES	280
330 - 520	DC	501	2240 - 2350	FK	306
530 - 1020	PU	375	2360 - 2470	UH	252
1030 - 1140	WW	374	2480 - 2590	XO	313
1150 - 1310	GW	266	2600 - 2710	TQ	332
1380 - 1510	LM	334	2720 - 2830	DI	327
1520 - 1630	AE	143	2840 - 2950	BG	276
1640 - 1750	EY	280	2960 - 5080	JN	444
1760 - 1870	OW	519	5090 - 5170	WX	515
1880 - 1990	AM	308	5180 - 5260	UY	432



MAZE

SoftSide[®] ADVENTURE SERIES



Issue 45 Adventure: Jack The Ripper II

Did the evil Ripper ever really die? You are an inspector at Scotland yard. Can you stop the Ripper before he strikes again?

SoftSide Adventure Series ~~CV~~ ~~DV~~

What would you say to a program that asks, "What do you want to do?" Well, you might say, "GET APPLE" or "KILL SPIDER", because that's how the *SoftSide Adventure Series* works.

Each issue, the latest Adventure takes you to another world of fantasy, puzzles and thrills. Your first task — survival — can be daunting until you

figure out the *right* way to do it. Each adventure sets a fundamental goal before you. You might have to rescue a princess, or depose a wicked ruler. Along the way, dozens of subsidiary tasks beset you. You'll have to be ingenious and persevering, and your rewards will be great.

To "win" a fantasy/adventure game, you must solve the author's devious puzzles, and overcome the obstacles that confront you — whether they be dragons or desperadoes. Death, should it come, is transitory — just re-run the program to live again!

Express your wishes with one- or two-word commands, like "LOOK", "NE" (for northeast), or "GET FROG". Use "I" to get an inventory of your possessions. The introduction to each Adventure explains this more fully.

Winning an Adventure is hard enough without the additional requirement that you do it all at once. In recognition of this, the Disk Version of the *Soft-Side Adventure Series* now features two new commands: "SAVE GAME" and "LOAD GAME".

To start up the Adventure, just run the program called "INTRO", "INTRO/BAS", or "INTRO.BAS" on your disk, or select the Adventure from the DV menu. On cassette, the INTRO program is the one just before the Adventure.

The Adventure runs in any TRS-80® with at least 16K RAM (32K disk).



Here are the encrypted hints for *Mad House*, the Adventure in Issue 44.

Silver dollar: WL MLG KFG RM EVMWRMT NZXSRMV FMGRO TREVM
GL TVLITV DZHSRMTGLM LI UORKKVW YB OFXPB XSFXPB.

Top hat: WL MLG TREV GL NZTRXRZM FMGRO GIRXPB WRXPB
SZH YVVM WVZOG DRGS.

Tricky Dicky: WIVHH FK ORPV ZYV ORMXL0M.

Shakespeare: TREV SRN Z ULFI-OVZU XOLEVI.

Bunny: ORPVH XOLEVI.

To get into the garden: HLFMW XZM YIVZP TOZHH.

To pass a day: HOVVK LM BLFI YFMP.

To escape:

A. VMGVI OZFMWIB ILLN LM KRXP FK WZB.

B. SZEVI Z ORG UOZHSORTSG.

C. SRWV RM OZFMWIB YZT.

D. OLLP OZFMWIB; GSVI TVG ZMW DVZI GSV FMRULIN.

E. OVZEV YZT ZUGVI GIFXP HGLKH.

F. VCRG WLLI.

General Information

These are the standard procedures for the programs published by **SoftSide** Publications, Inc. Sometimes, a particular program does not lend itself to these procedures. Always read the specific instructions accompanying a program. They will instruct you if there are any variances from the following procedures. Also, back issues of **SoftSide** Magazine may differ in some details.



At the conclusion of each **SoftSide** listing, we include a **STOMP** Table. **STOMP** for the TRS-80 appeared in Issue #45. **STOMP**

supersedes **SWAT** as **SoftSide's** standard debugging tool to help those who type BASIC programs from the pages of **SoftSide Selections**. If you don't have **STOMP**, we'll send you a free reprint. Send a business-sized, self-addressed, stamped envelope to:

SoftSide Publications, Inc.
Department **STOMP**
10 Northern Blvd.
Northwood Executive Park
Amherst, NH 03031

Be sure to tell us that you have a TRS-80 computer.

Magnetic Media

Disks are available in Model I or Model III format. They contain the DOSPLUS operating system. A cover program runs automatically when you boot the disk. To avoid entering BASIC and running the cover program, press and hold the Enter key as the disk boots.

Back issues prior to May, 1981, are available in Model I format only. If you have a two-drive Model III, you can convert such disks with the Convert utility.

Tapes CLOAD in the normal manner on Model I's, and at low speed (500 baud) on Model III's. The first program is a cover program; side two of the tape duplicates side one.

SoftSide Selections disks and tapes are duplicated on reliable, professional equipment. Bad copies are exceedingly rare. Nevertheless, the trip through the mail occasionally results in damage to the sensitive magnetic media. If, after a reasonable number of tries on clean, well-adjusted equipment, you are unable to load a program, return it to us with a precise explanation of the problem. We shall send you a replacement copy.

SoftSide Selections media are not copy-protected. We urge you to make an archival backup copy of your disk or tape as soon as you receive it, as our replacement policy is valid only for 30 days. Please resist the urge to give away copies of copyrighted material.

Line Listings

The line listings in this booklet are in standard 64-column format, and they appear exactly as they should on your screen when you type LIST. Because **STOMP** ignores spaces between keywords, you may add them to enhance the legibility of the program you are typing. You should note, however, that each additional space occupies one more byte of valuable memory. If you are not sure about the availability of memory, add no additional spaces.

- Lower-case characters: Some TRS-80s have lower case. Many programs we publish contain lower case. Since **STOMP**, **SoftSide's** debugging utility, ignores the case of characters, you may use capital letters if you so desire or if your computer doesn't have lower case.

Copyright © 1983 SoftSide Publications, Inc.

These programs are for your personal use only. Please resist the urge to give away copies of copyrighted material.



Tired of Typing?

- Use And Enjoy These Programs Right Away!
- Avoid Worry About Typos!
- Get Additional Programs to Enjoy Without Typing.



Order this Issue's Programs on Disk or Cassette!

Send the coupon below to:

SoftSide Publications, Inc.
10 Northern Blvd.
Northwood Executive Park
Amherst, NH 03031



SoftSide

CV

SoftSide

DV

SoftSide® Selections

ORDER THIS ISSUE'S MEDIA

Having difficulty finding the time to input the programs from this issue of **SoftSide Selections**? Don't despair — the cassette and disk versions are still available! **And**, each DV (disk version) contains an additional program.

☐ #45 DV \$16.00 ☐ #45 CV \$8.00

☐ TRS-80® Mod. I ☐ TRS-80® Mod. III

Name _____

Address _____

City/State _____ Zip _____

☐ **Check or Money Order**

☐ MasterCard ☐ VISA

Name of Cardholder _____

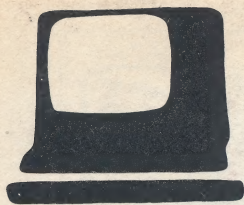
MC #/VISA # _____

Exp. Date ____/____/____ Signature _____

SoftSide® Selections



A product of SoftSide Publications, Inc.
10 Northern Blvd., Amherst, NH 03031



SoftSide[®] Selections

Here's **SoftSide Selections**, the handy, pull-out booklet with program listings for your TRS-80 Model I or III computer. This issue, **SoftSide Selections** for the TRS-80 features:

- **Sea Base** puts you in charge of an undersea city with a waning geothermal energy source.

- **Maze** — This program puts you inside a three-dimensional maze.

- **TRS-80 DV Bonus Program: Filter**

Use this power print formatting program, in conjunction with your word processor, to control your printer and format your text, according to your specifications.

- **The SoftSide Adventure Series —**

Jack The Ripper II, by Peter Kirsch

Your grandfather chased the evil Ripper to a watery death in the River Thames, or so he thought. Now it seems that the Ripper lives again...

SOFTSIDE PUBLICATIONS, INC.

Amherst, New Hampshire